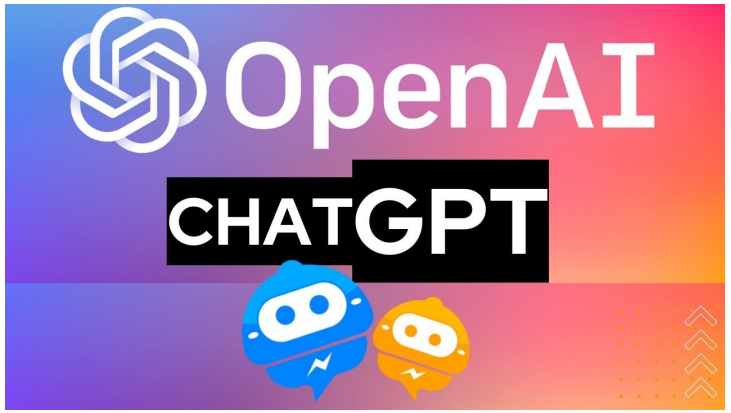
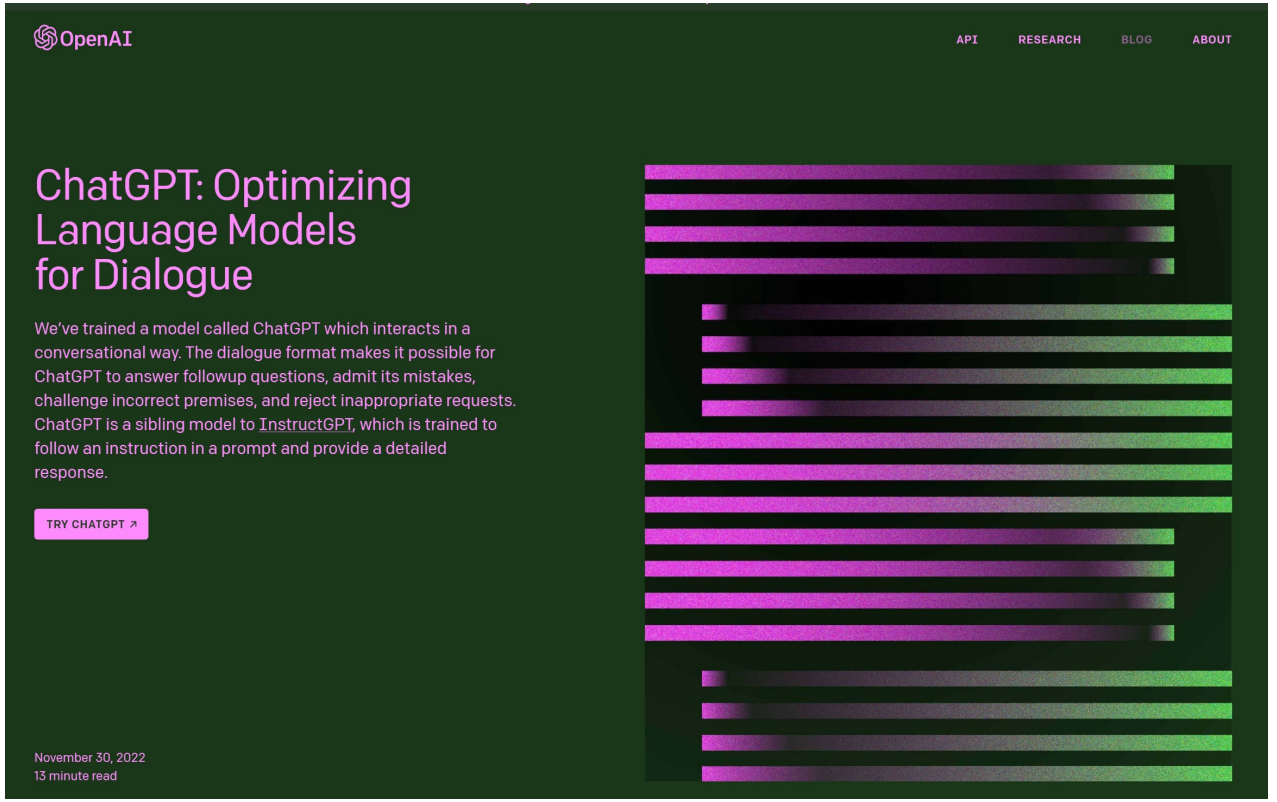
迈向通用长程智能体：挑战与创新

复旦大学
黄萱菁

大模型引领人工智能的范式革命

2022 年11 月30 日，OpenAI 发布了AI 对话模型，被认为是人工智能里程碑式应用





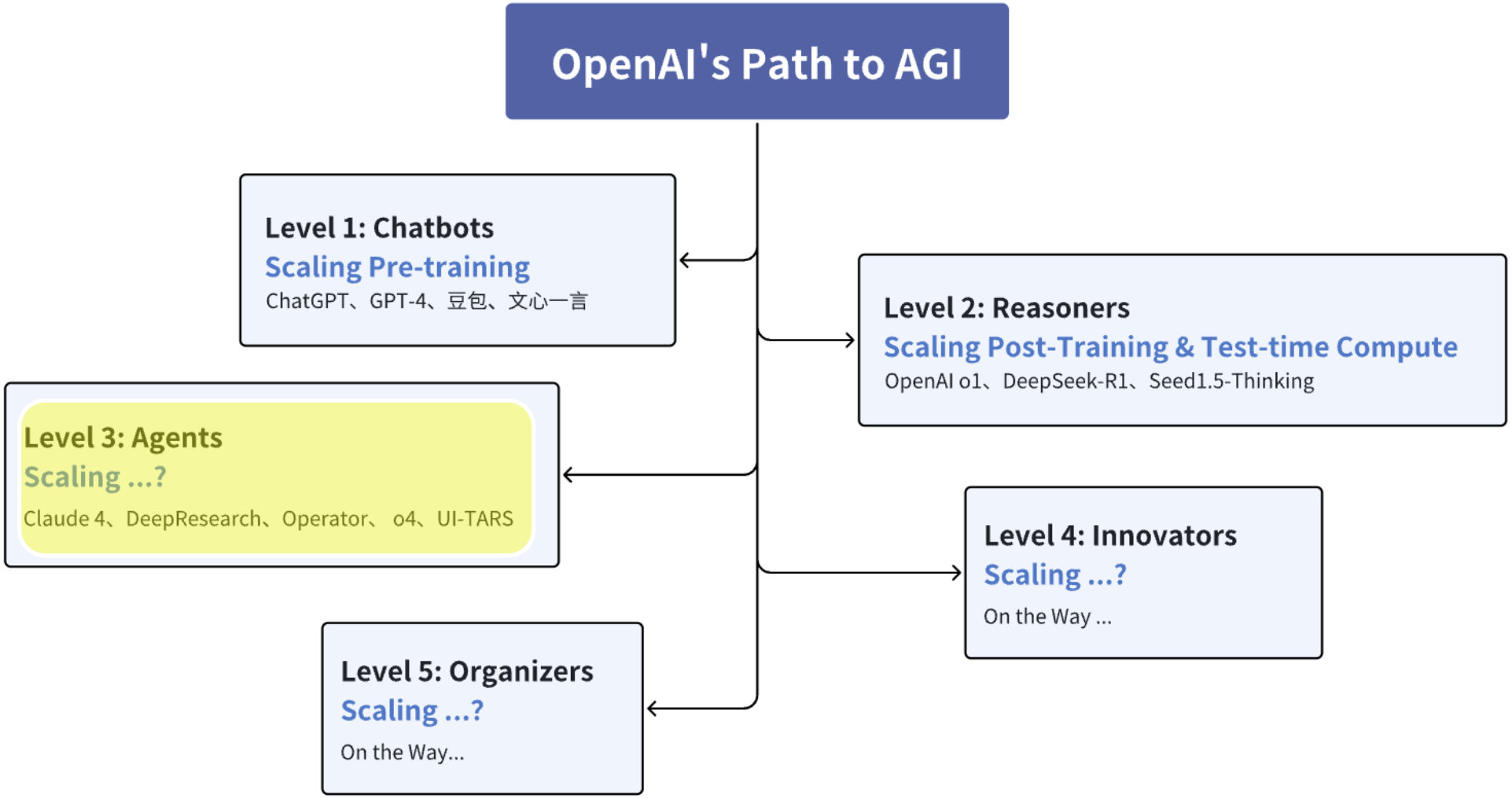
像ChatGPT这样的AI聊天机器人将变得与个人电脑或互联网同样重要

“ChatGPT是1980年以来最具革命性的科技进步”

什么是大模型(LLM)

- 大模型是指具有**大规模参数**和**复杂计算结构**的机器学习模型
- 当模型的训练数据和参数达到一定的临界规模后，其表现出了一些**先前无法预测的、更复杂的能力和特性**，模型能够从原始训练数据中自动学习并发现新的、更高层次的特征和模式，这种能力被称为“**涌现能力**”。
- 相比一般模型，大模型通常具有参数量大、层数深的特点，具有更强的表达能力和更高的准确度，但也**需要更多的计算资源和时间来训练和推理**，适用于数据量较大、计算资源充足的场景，例如云端计算、高性能计算、人工智能等。

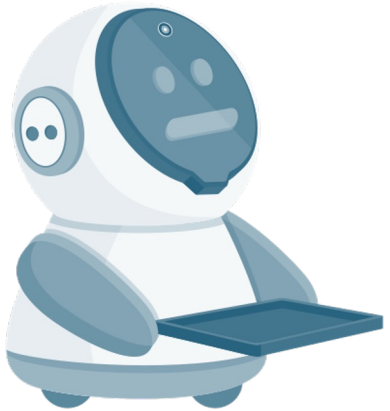
从大模型到通用人工智能(AGI)



什么是智能体（AI Agent）？

AI Agent:

能够通过感知器感知周围环境，做出决策，然后通过执行器采取行动的人工智能实体



感知周围环境

↓

规划与决策

↓

采取行动

[1] Wooldridge, M. J., N. R. Jennings. Intelligent agents: theory and practice. Knowl. Eng. Rev., 10(2):115–152, 1995.
[2] The Rise and Potential of Large Language Model Based Agents: A Survey. 2309.07864, 2023.

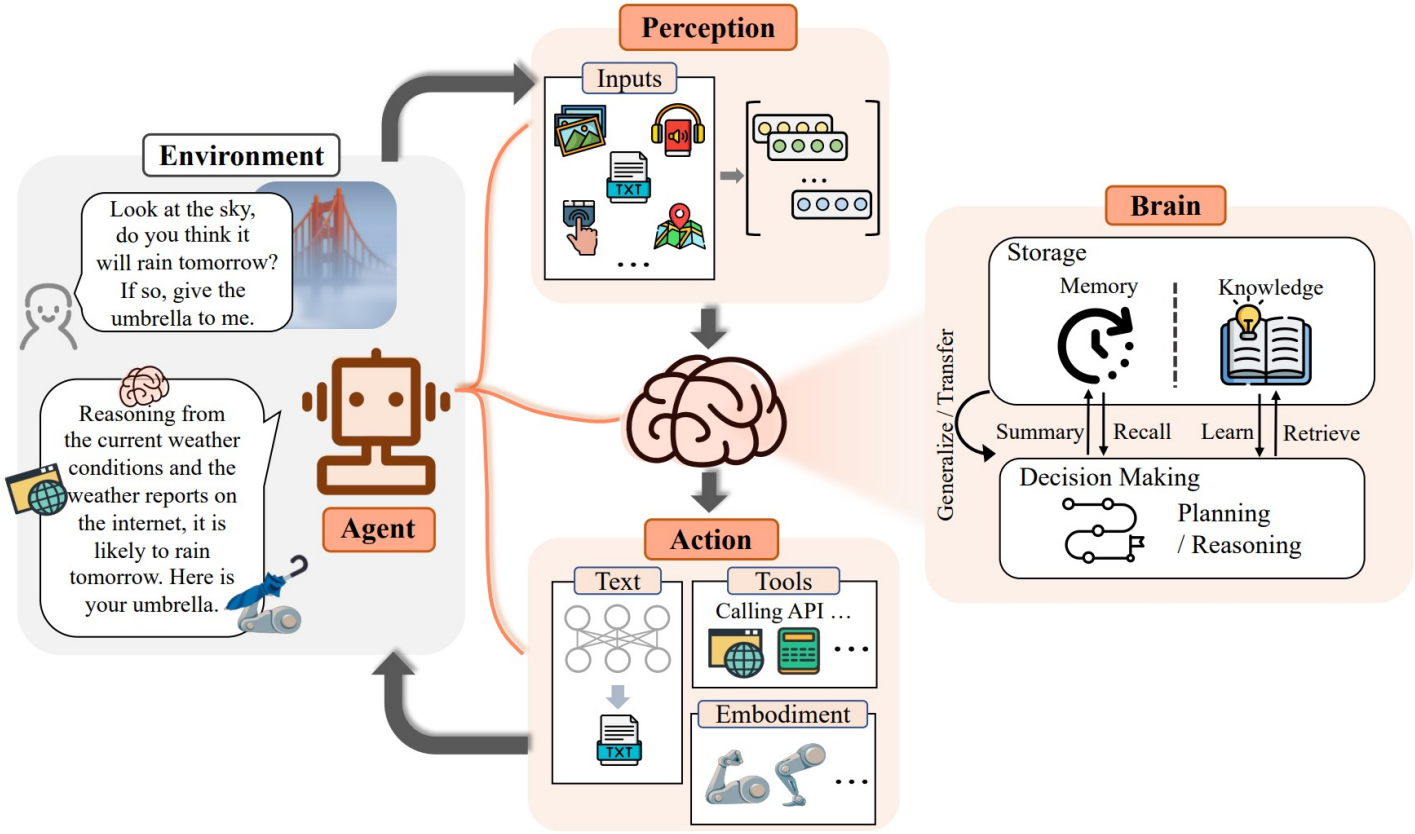
什么是基于大语言模型的智能体？ (LLM-based Agent)

LLM-based Agent:

以大语言模型（LLM）作为核心控制器的 智能体，并通过用于感知、决策和行动的各种组件实现增强

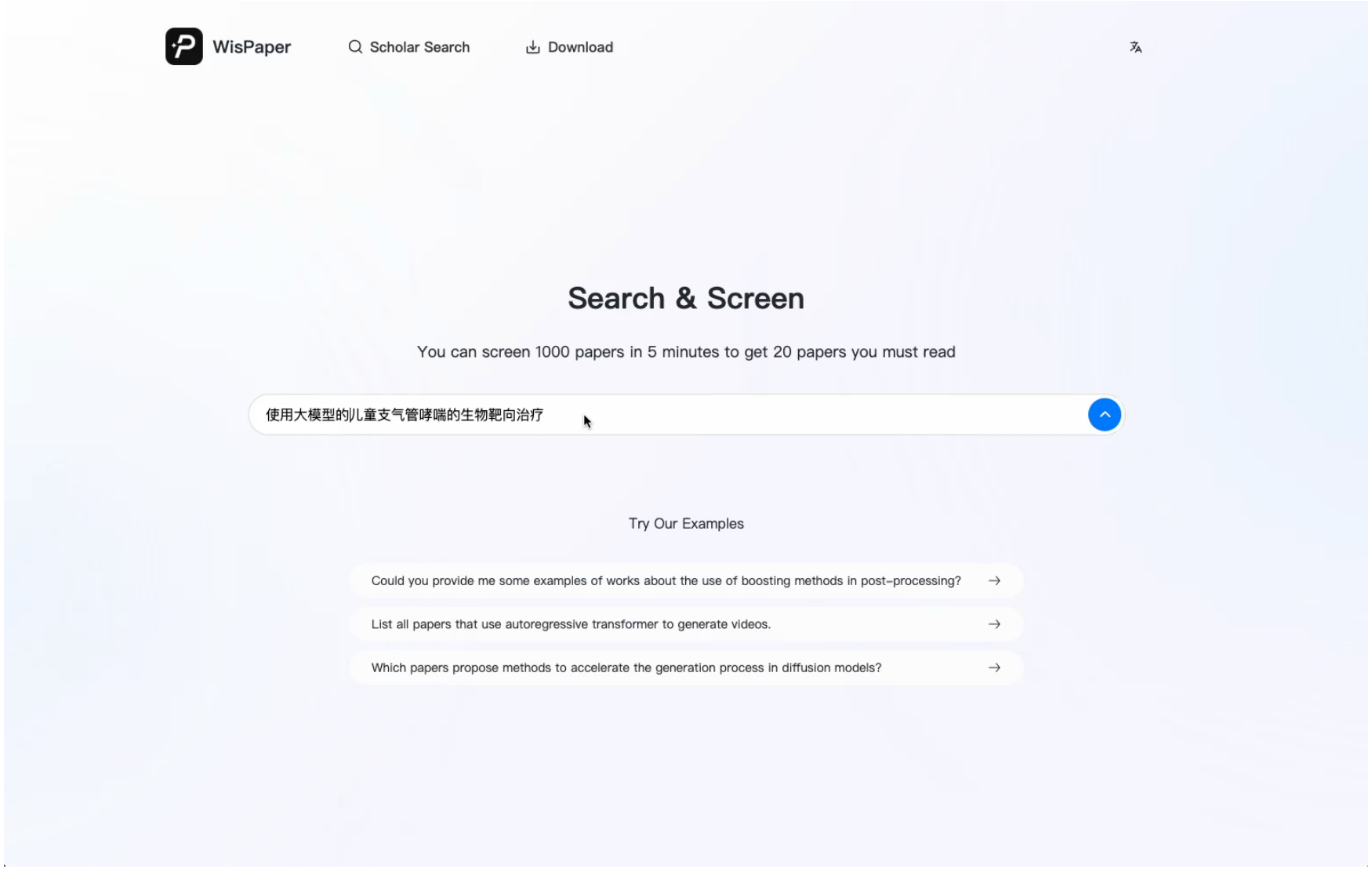
能力：

- 感知与理解能力
  Perceiving & Understanding
- 推理与规划能力
  Reasoning & Planning
- 记忆存储与检索能力
  Memory Storing & Retrival
- 生成与行动能力
  Generation & Action

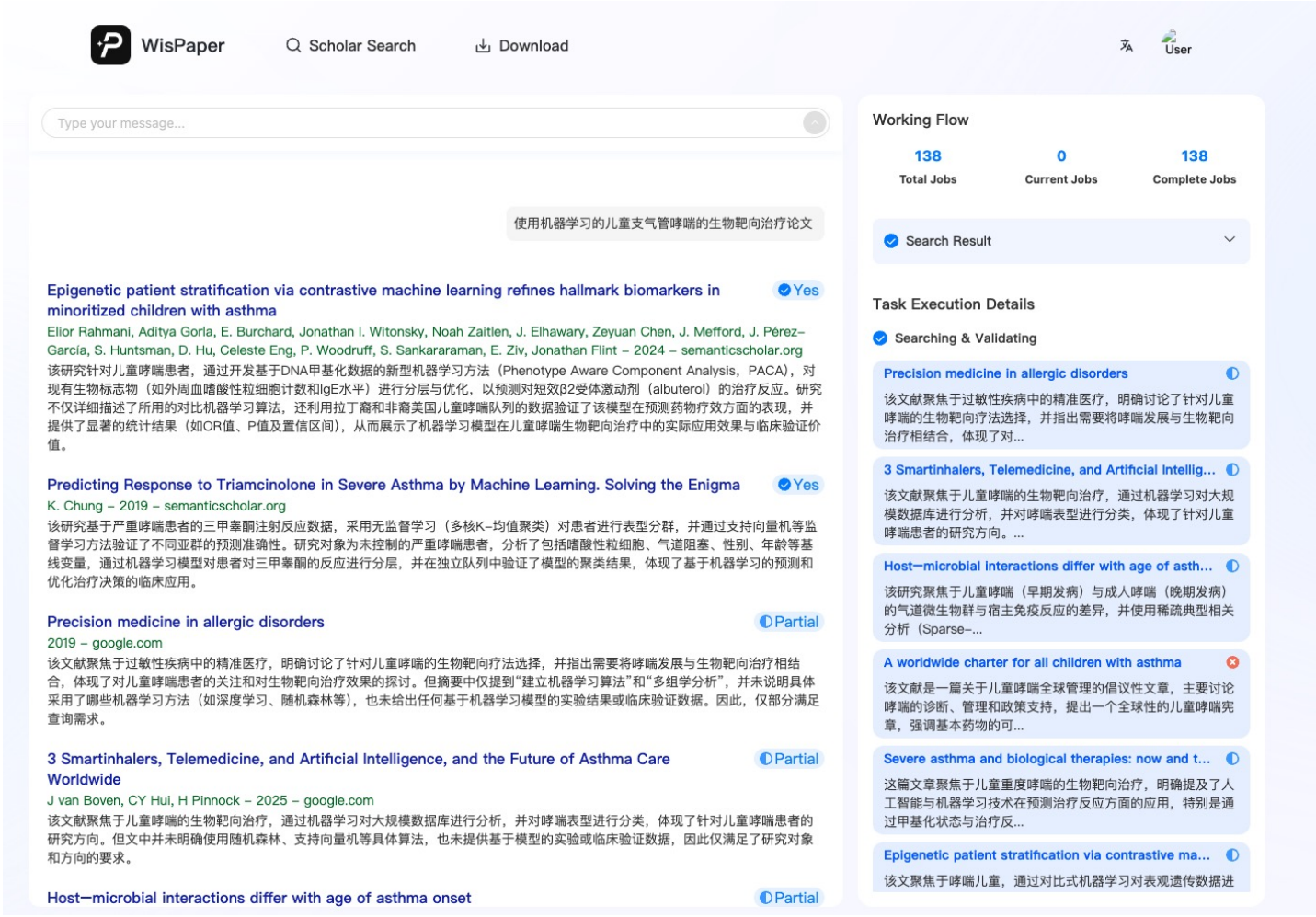


[1] The Rise and Potential of Large Language Model Based Agents: A Survey. 2309.07864, 2023.
[2] Cognitive Architectures for Language Agents. TMLR, 2024.

WisPaper： AI科研助手



WisPaper：AI科研助手



NEX： 论文海报生成

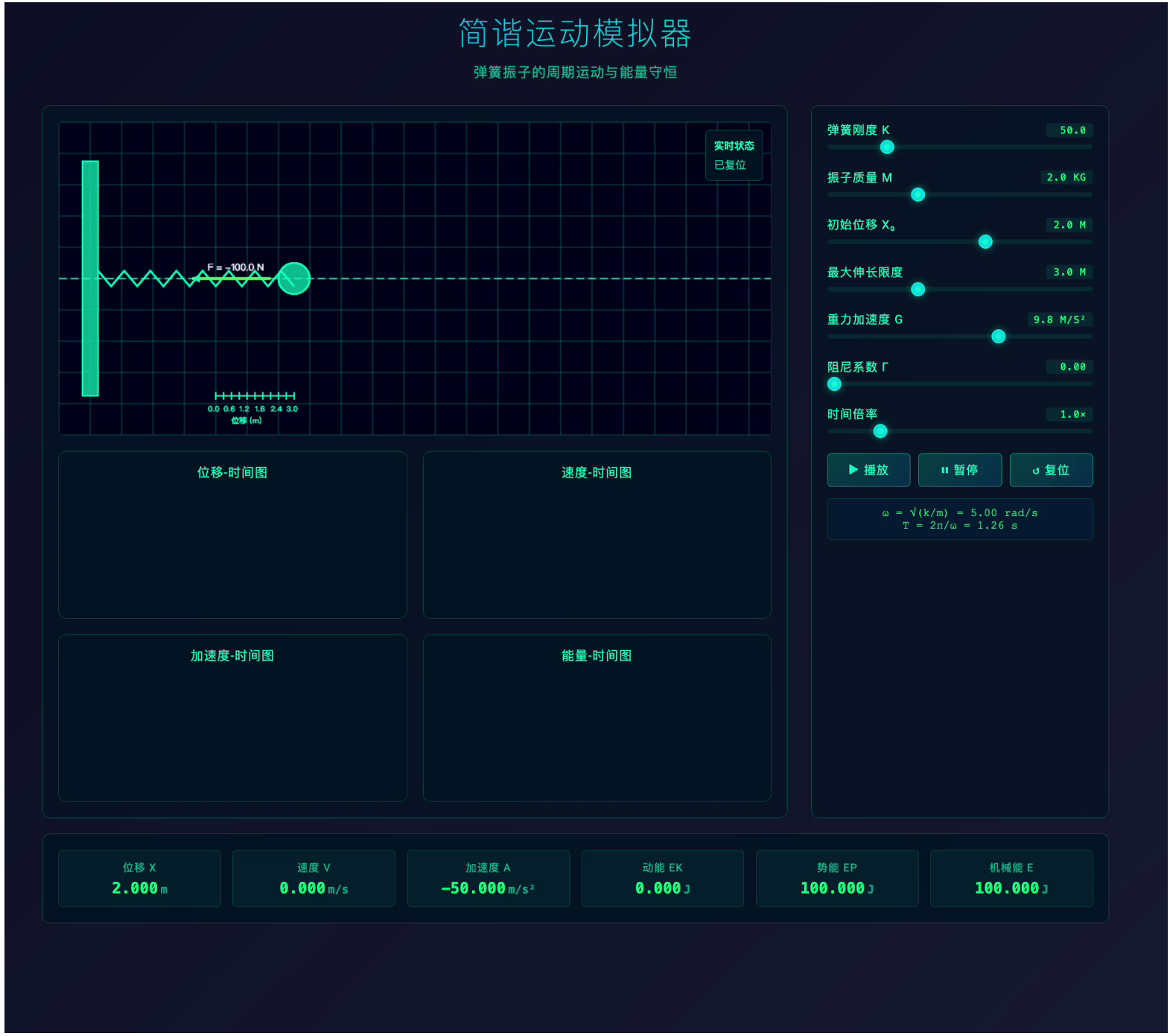
[Query] 帮我从uploads/2510.18927v1.pdf生成一张海报

NEX： 课件制作

[Query]

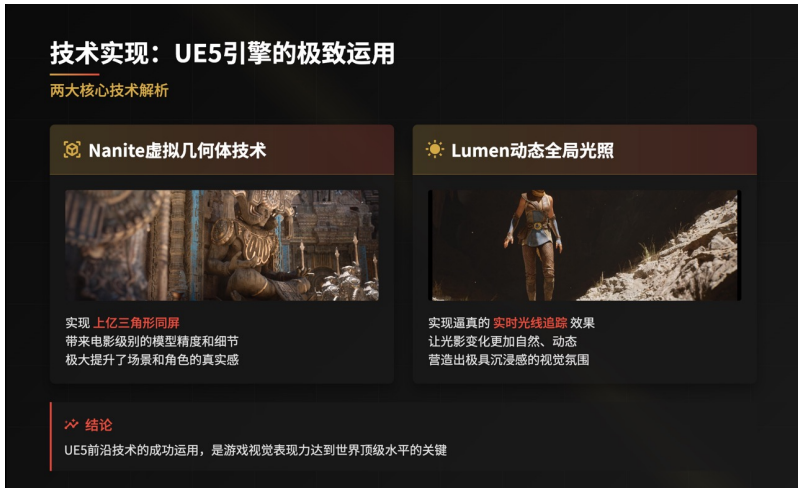
我需要一个模拟小球连着弹簧的运动动画。具体要求如下。

1. **物理模型**：小球一端连接弹簧，弹簧另一端固定在墙上；
2. **参数可调**：可以调整弹簧的刚度系数k，以及允许的最大伸长限度；
3. **初始条件**：模拟弹簧从平衡位置被拉伸n米后释放；
4. 动画能通过参数修改来演示不同k值和最大伸长下的运动情况



NEX：深度调研+ppt制作

[Query] 请深度调研分析《黑神话悟空》的成功，包括研发、发布、评价等全方面，并制作一份幻灯片报告讲解。



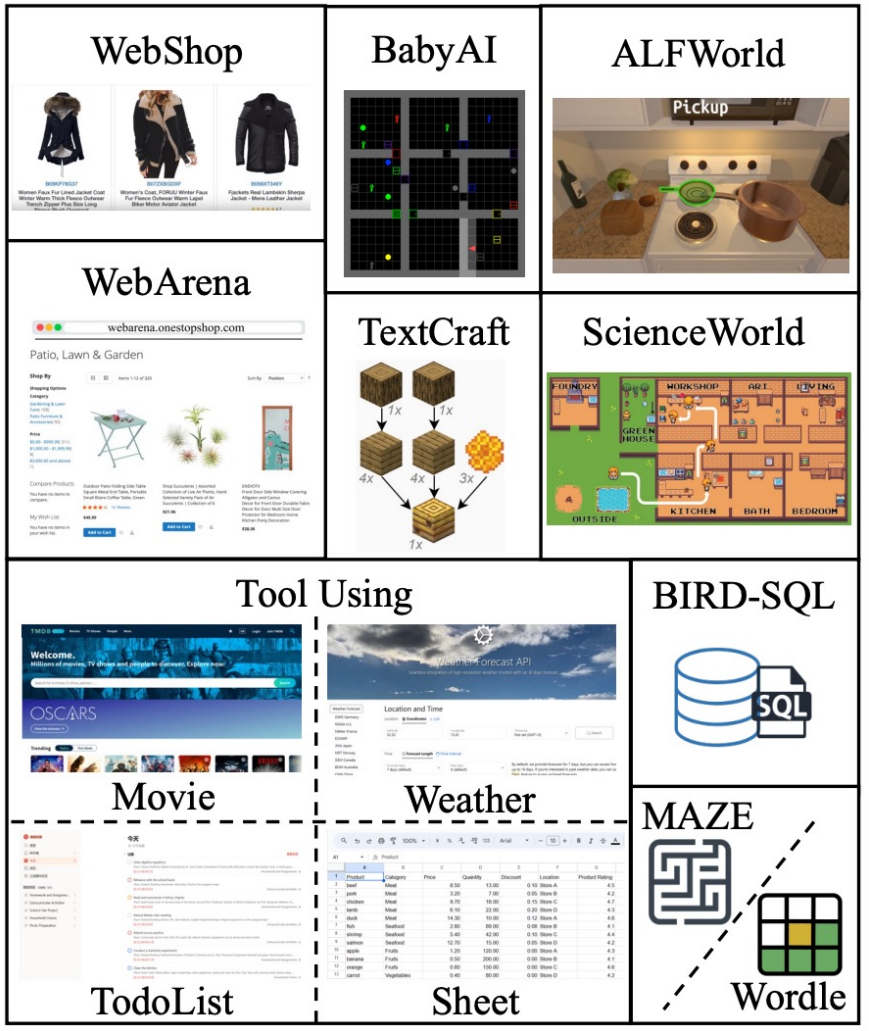
大模型智能体的使命与任务

任务：

自主解决模拟世界或现实世界中时间跨度长、复杂性高的交互式决策任务，例如网页任务、具身任务、工具使用任务等

LLM v.s. LLM-based Agent:

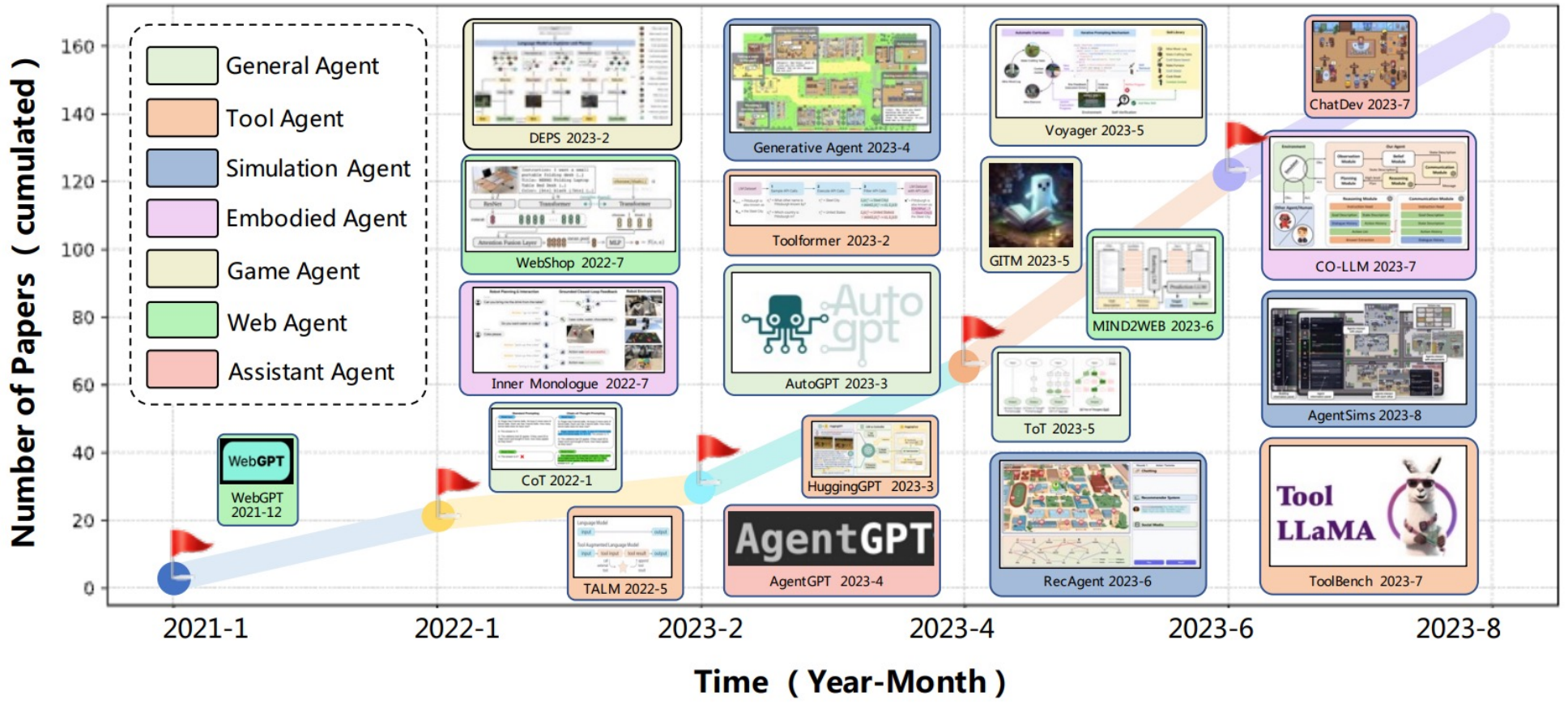
- 聊天机器人 v.s. 人类助理
- 简单答疑 v.s. 决策制定



[1] The Rise and Potential of Large Language Model Based Agents: A Survey. ArXiv 2309.07864, 2023.
[2] AgentGym: Evolving Large Language Model-based Agents across Diverse Environments. ArXiv 2406.04151, 2024.

基于大语言模型的智能体的发展

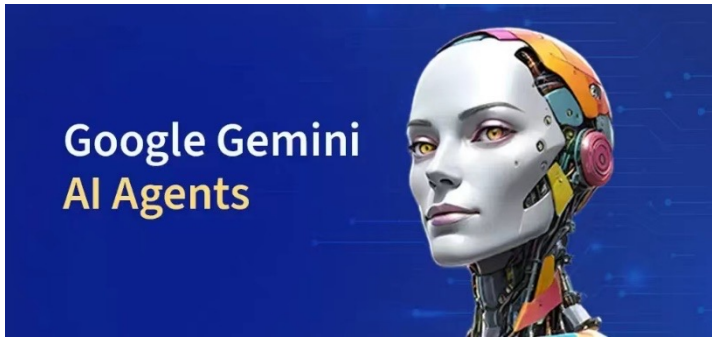
- 智能体：通向通用人工智能的一条有前景的道路
- 通过专业智能体的深度连接，人工智能将带来服务的代际升级

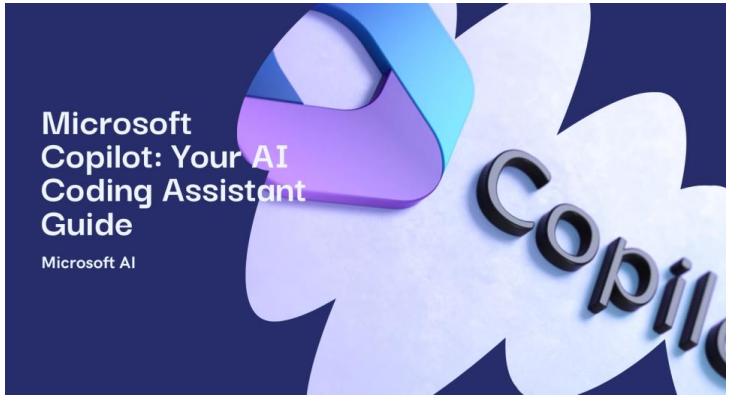


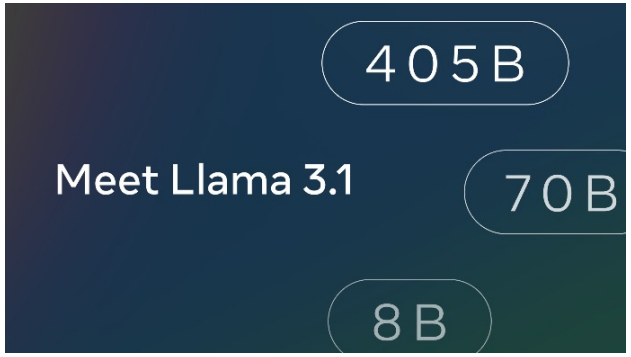
[1] A Survey on Large Language Model based Autonomous Agents. 2308.11432, 2023.

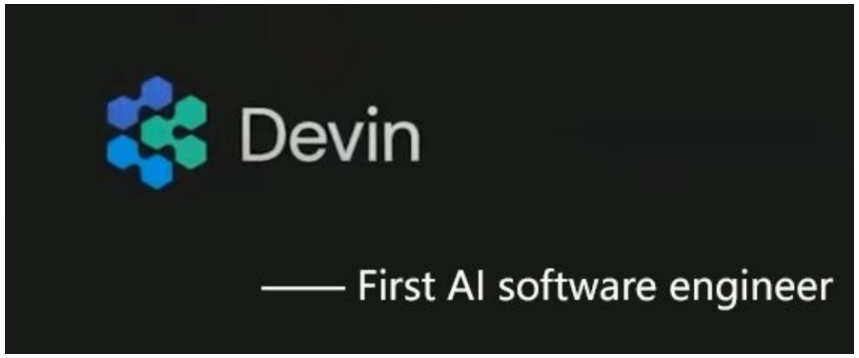
基于大语言模型的智能体的发展

- **工业界**的广泛关注
- OpenAI, Microsoft, Meta, Google…









基于大语言模型的智能体的发展

- 大量智能体开源项目

Auto-GPT[1]

截至目前，已获得超 **166k stars**

Standford Smallville[3]

第一个模拟可信人类行为的**智能体环境**

LangChain[2]

截至目前，已获超 **91k stars**

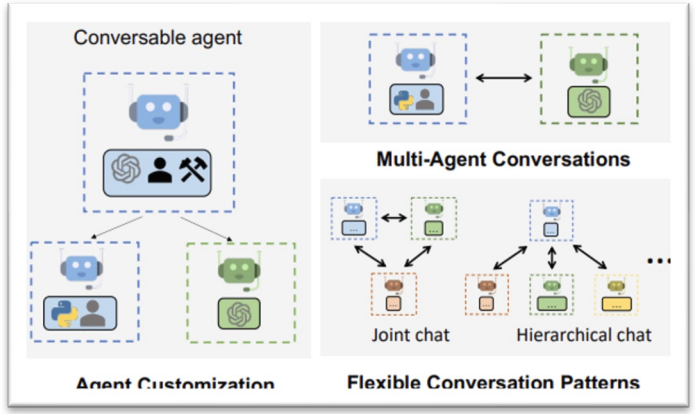
Agent Survey[4]

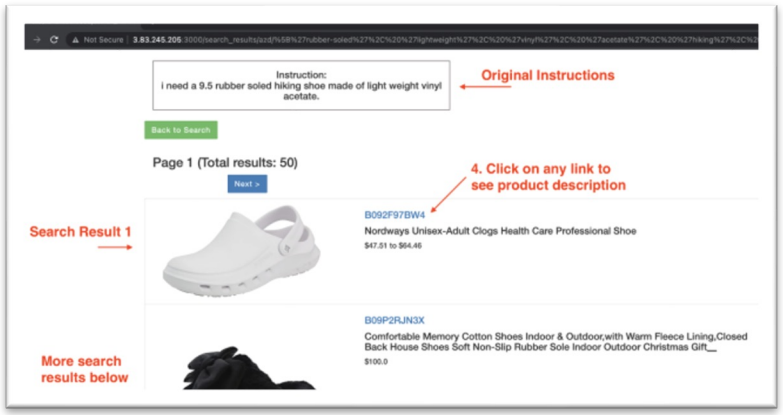
截至目前，已获得超 **6.1k stars**

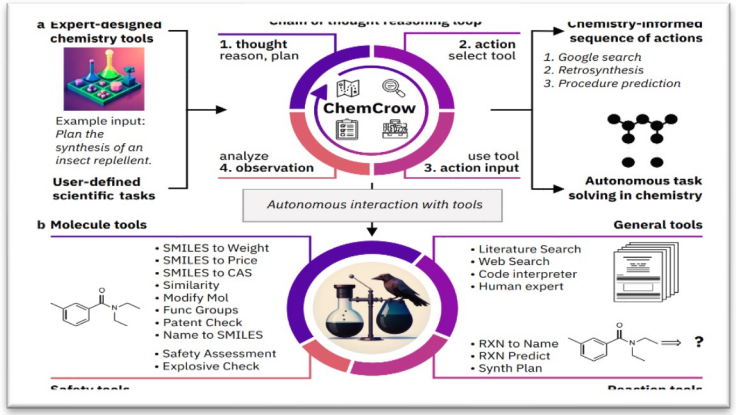
[1] Gravitas, S. Auto-GPT: An Autonomous GPT-4 experiment, 2023. URL https://github.com/Significant-Gravitas/Auto-GPT, 2023.
[2] Chase, H. LangChain. URL https://github.com/hwchase17/langchain, 2022.
[3] Park, Joon Sung, et al. Generative agents: Interactive simulacra of human behavior. Proceedings of the 36th annual acm symposium on user interface software and technology. 2023.
[4] The Rise and Potential of Large Language Model Based Agents: A Survey. 2309.07864, 2023.

基于大语言模型的智能体的发展

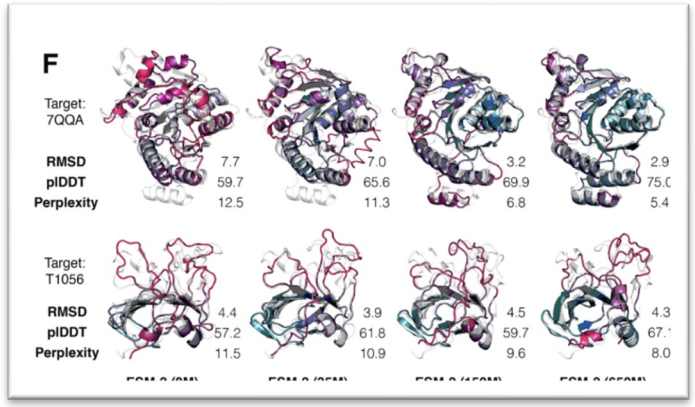
- 各种场景下的广泛**应用**

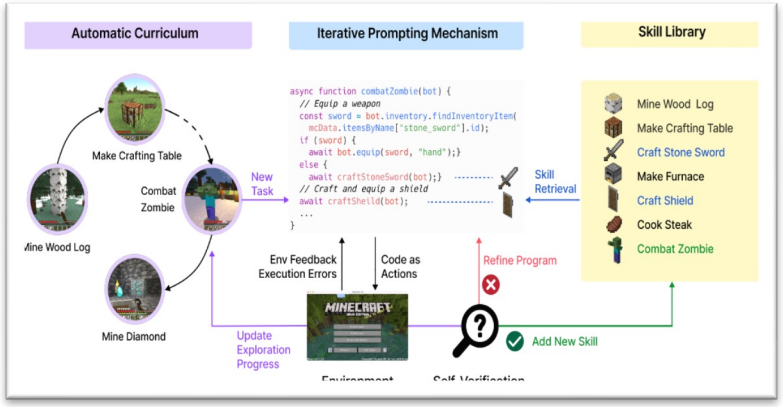


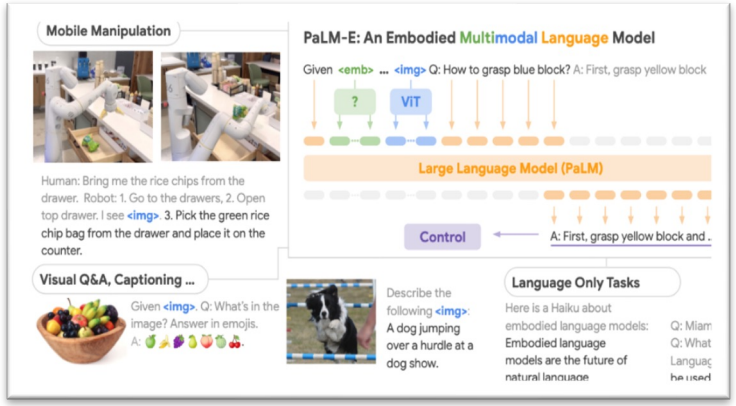




聊天 / 家居 / 网页 / 编程 / 医疗 / 生物 / 游戏 / 社会 / 具身 … …





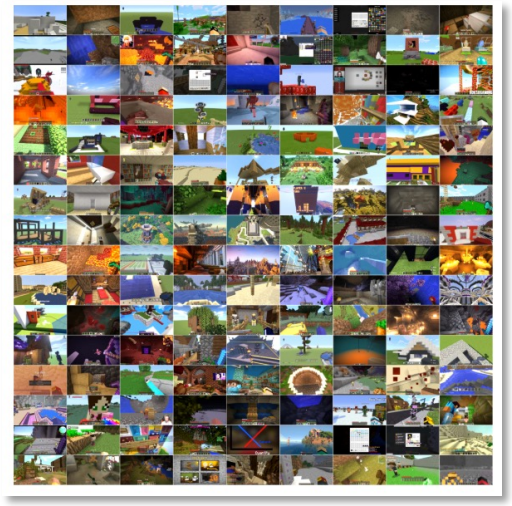


[1] Wu, Qingyun, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155 (2023).
[2] Yao, Shunyu, et al. Webshop: Towards scalable real-world web interaction with grounded language agents. NeurIPS, 2022.
[3] Bran et al. Chemcrow: Augmenting large-language models with chemistry tools, 2023.
[4] Lin, Zeming, et al. "Evolutionary-scale prediction of atomic-level protein structure with a language model." Science 379.6637 (2023): 1123-1130.
[5] Raad, Maria Abi, et al. Scaling Instructable Agents Across Many Simulated Worlds. arXiv preprint arXiv:2404.10179 (2024).
[6] Driess D, Xia F, Sajjadi M S M, et al. Palm-e: An embodied multimodal language model[J]. arXiv preprint arXiv:2303.03378, 2023.

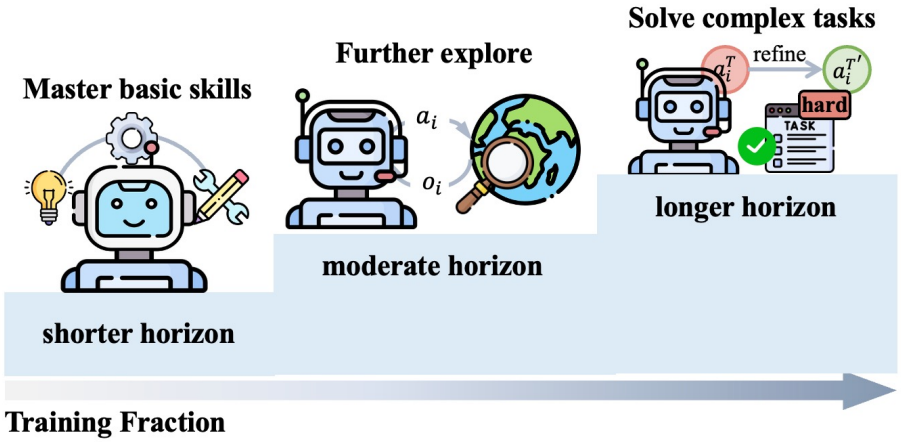
大模型智能体面临挑战——Scaling Challenges

在这个时代，构建通用的大模型智能体，我们应该 Scale Up 什么？

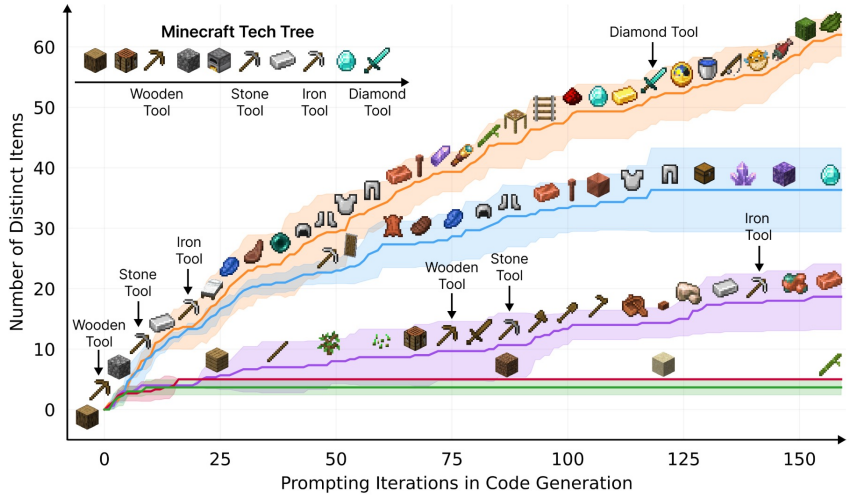
Scaling Environment



Scaling Interaction



Scaling Goals



[1] Xi et al., AgentGym-RL: Training LLM Agents for Long-Horizon Decision Making through Multi-Turn Reinforcement Learning.
[2] Fan et al., MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. NeurIPS 2022.
[3] Wang et al., Voyager: An Open-Ended Embodied Agent with Large Language Models. TMLR.

为什么要扩展环境

环境是智能体能力的“上限”：没有多样、真实、高质量的环境，就学不出可泛化的策略。

Environments Hub（Karpathy）：统一接口、开放共享，降低环境构建门槛，鼓励社区共建与复用。

Google SIMA：以“多世界、多任务、可指令化”为核心，用真实/游戏环境 + 人类评测形成训练闭环。

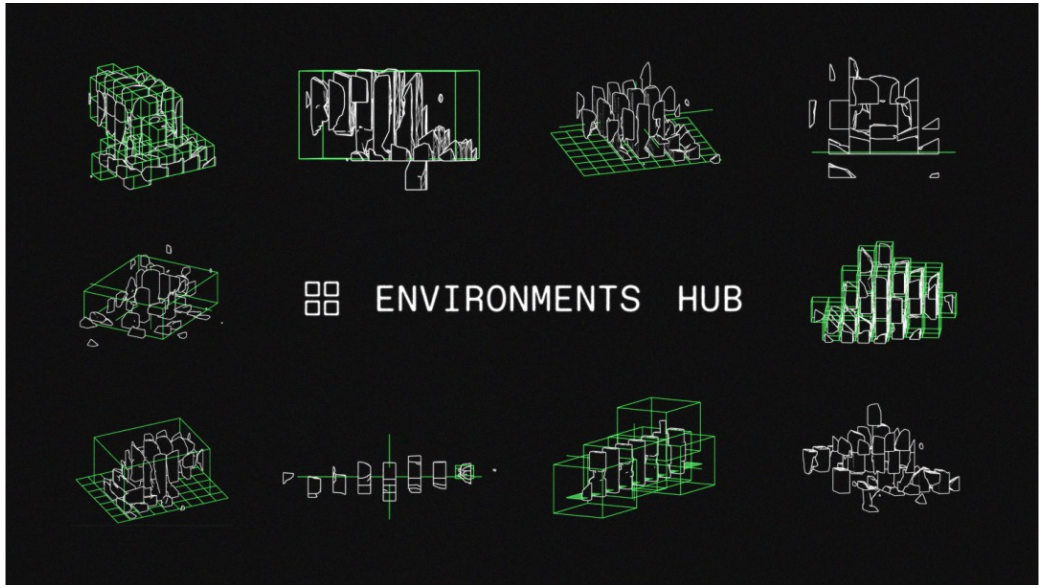
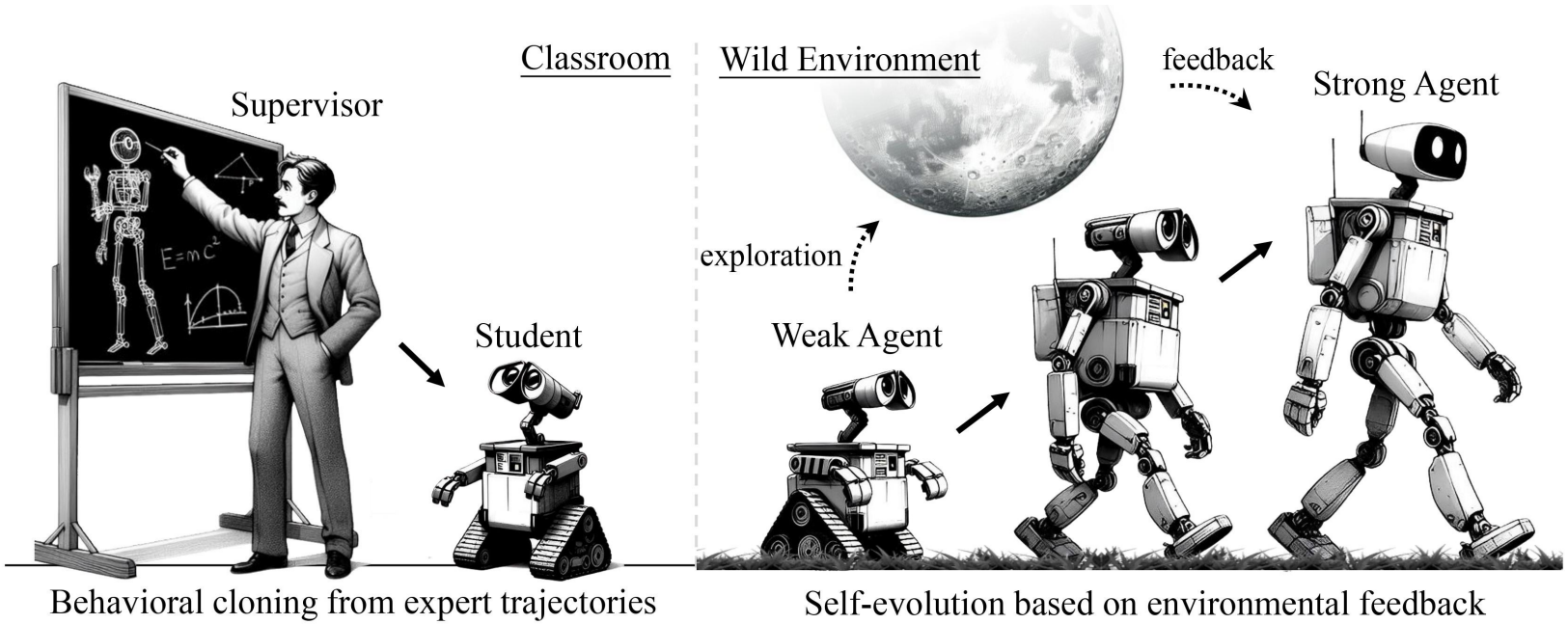


Figure 1 | **Overview of SIMA.** In SIMA, we collect a large and diverse dataset of gameplay from both curated research environments and commercial video games. This dataset is used to train agents to follow open-ended language instructions via pixel inputs and keyboard-and-mouse action outputs. Agents are then evaluated in terms of their behavior across a broad range of skills.

我们的探索：AgentGym



跨环境的自我进化:

面对环境碎片化和任务静态化，我们希望构建一种能够跨环境持续学习的智能体体系。

AGENTGYM: Evolving Large Language Model-based Agents across Diverse Environments

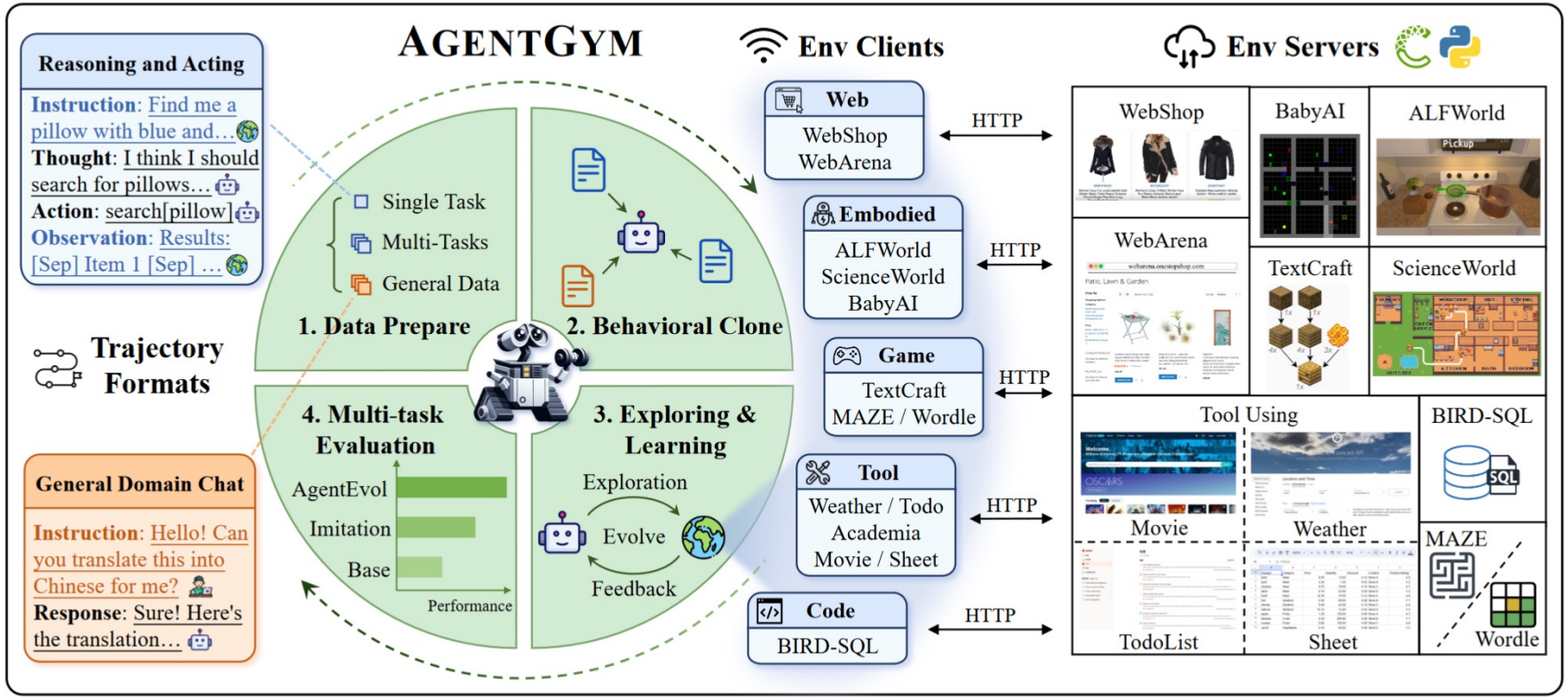
Zhiheng Xi*†, Yiwen Ding*, Wenxiang Chen*,
Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao,

Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou,
Tao Gui†, Qi Zhang†, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang

Fudan NLP Lab & Fudan Vision and Learning Lab

AgentGym: 交互平台，评估基准，轨迹数据集

- 丰富的环境支持: 14 个智能体环境与 89 个不同任务
- 统一的交互接口: 由HTTP服务进行部署



基准测试套件与轨迹集

- AgentInstrution: 使用self-instruct进行轨迹扩增
- AgentEval: 一个多样且具有挑战性的评估子集
- AgentTraj: ReAct 格式, 使用环境奖励值进行过滤

Env.	Task Num	Instr.	Eval.	Traj	Traj-L	Rounds
WA	3	812	20	0	0	–
WS	1	6910	200	1000	3930	5.1
MZ	1	240	25	100	215	4.3
WD	1	980	25	500	955	4.3
ALF	6	3827	200	500	2420	13.3
Sci	30	2320	200	1000	2120	19.9
Baby	40	900	90	400	810	5.7
TC	1	544	100	300	374	8.0
WT	1	343	20	160	311	5.5
MV	1	238	20	100	215	4.0
AM	1	20	20	0	0	–
ST	1	20	20	0	0	–
TL	1	155	20	70	135	5.6
BD	1	3200	200	2000	3000	1.0
Total	89	20509	1160	6130	14485	–

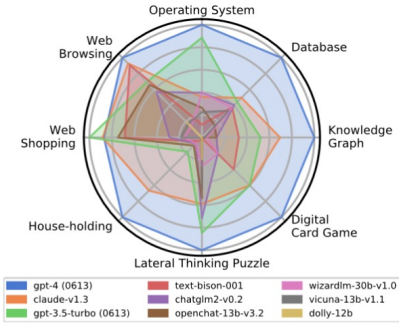
AgentGym vs. 其他agent框架

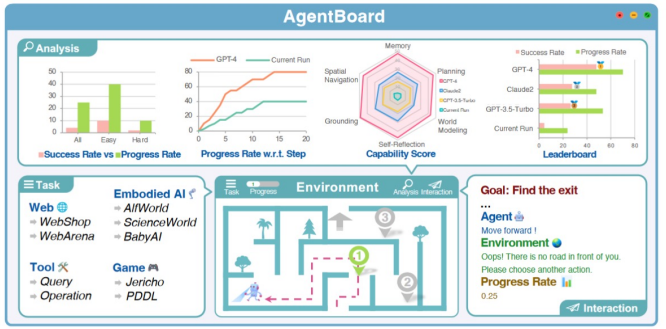
不仅仅是评测基准和训练流程！

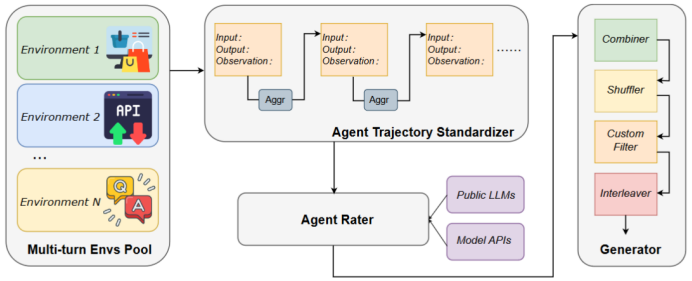
交互式平台

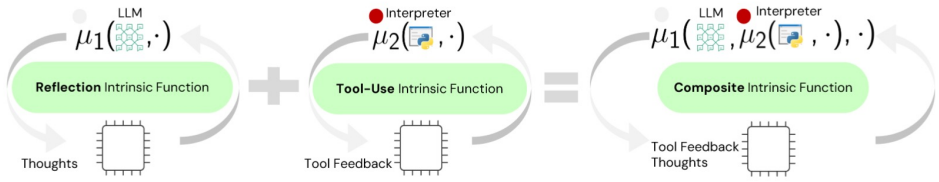
- 集成更多环境
- 收集了开源的轨迹集
- 训练与自我进化范式

Frameworks	Env.	Inter. Plat.	Traj.	Evol.
AgentBench [36]	8	Eval	No	No
AgentBoard [31]	12	Eval	No	No
AgentOhana [18]	10	No	Yes	No
Pangu-Agent [37]	6	No	Yes	Single-Env
AGENTGYM (Ours)	14	Eval & Train	Yes	Multi-Env









[1] AgentBench: Evaluating LLMs as Agents. ArXiv 2308.03688, 2023.
[2] AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. ICLR 2024.
[3] AgentOhana: Design Unified Data and Training Pipeline for Effective Agent Learning. ICLR 2024.
[4] Pangu-Agent: A Fine-Tunable Generalist Agent with Structured Reasoning. ArXiv 2312.14878, 2023.

AgentEvol : Self-Improve in Environment

使用收集的轨迹进行行为克隆

- 通用能力智能体的基础

Exploration + Learning 进化循环

- **Exploration Step**:
- 在之前**未见过**的任务和指令上进行探索
- **Learning Step**:
- 基于反馈使用奖励加权损失对初始智能体进行微调和优化

Algorithm 1: AGENTEVOL

Input: Initialized policy LLM-based agent π_θ, environment set $\mathcal{E}$, trajectory subset $\mathcal{D}_s$, full instruction set $\mathcal{Q}$, reward function r.

Procedure Behavioral cloning:

Maximize objective $\mathcal{J}_{BC}(\theta) = \mathbb{E}_{(e,u,\tau)\sim\mathcal{D}_s}\Big[\log \pi_\theta(\tau|e,u)\Big]$ to get $\pi_{\theta_{base}}$;

Procedure Evolution :

$\pi_{\theta^1} \leftarrow \pi_{\theta_{base}}$;

for iteration $m = 1$ to M **do**

// Perform **Exploration Step**

$\mathcal{D}_m = \bigcup_{e\in\mathcal{E}} \mathcal{D}_m^e$, where $\mathcal{D}_m^e = \{(e, u^j, \tau^j) \,|u^j \sim \mathcal{Q}_e, \tau^j \sim \pi_{\theta^m}(\tau|e,u^j)\}_{j=1}^{|\mathcal{D}_m^e|}$;

Compute reward for $\mathcal{D}_m$ with r;

$\mathcal{D}_m \leftarrow \mathcal{D}_m \cup \mathcal{D}_s$;

// Perform **Learning Step**

Maximize objective $\mathcal{J}_{Evol}(\theta) = \mathbb{E}_{(e,u,\tau)\sim\mathcal{D}_m}[r(e,u,\tau)\log \pi_\theta(\tau|e,u)]$ to get $\pi_{\theta^{m+1}}$;

end

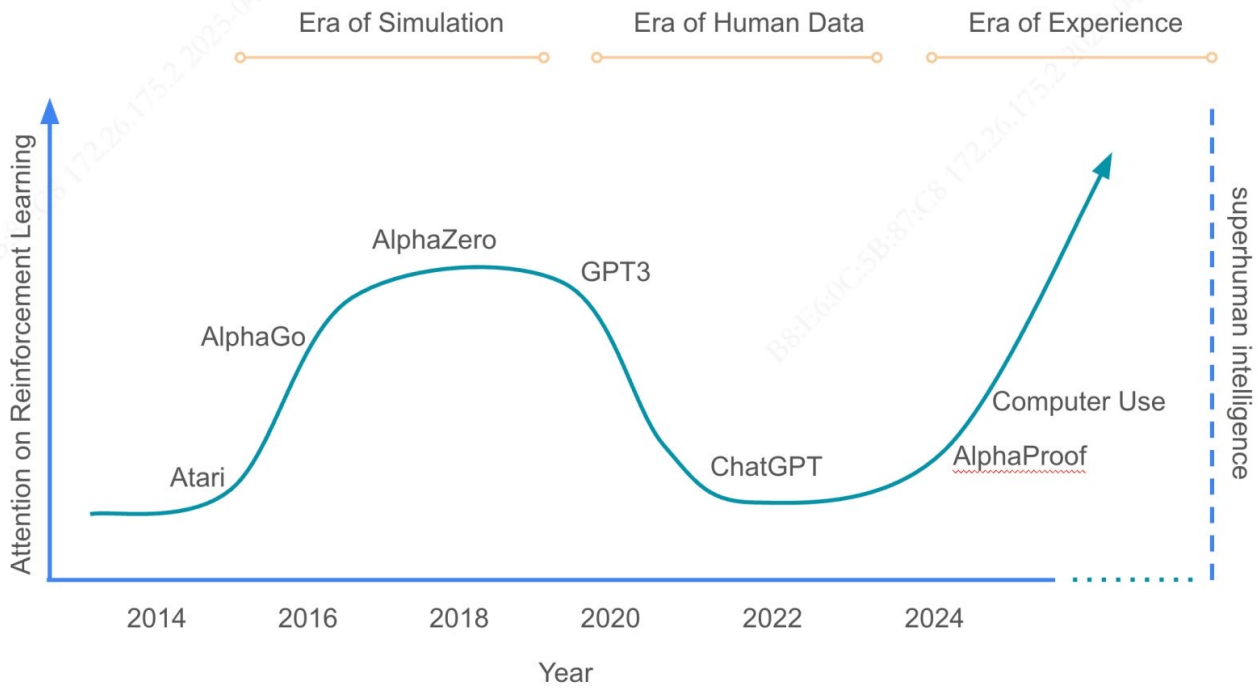
跨环境实验结果

在多样化任务上的评测结果

- 从有限的轨迹开始模仿
- 探索更广泛的区域

Method	WS	ALF	TC	Sci	Baby	MZ	WD	WT	MV	TL	BD
Close-sourced Models & Agents											
DeepSeek-Chat	11.00	51.00	23.00	**16.80**	45.67	4.00	24.00	70.00	70.00	75.00	13.50
Claude-3-Haiku	5.50	0.00	0.00	0.83	1.93	4.00	16.00	55.00	50.00	65.00	13.50
Claude-3-Sonnet	1.50	13.00	38.00	2.78	**79.25**	0.00	36.00	65.00	80.00	80.00	**17.00**
GPT-3.5-Turbo	12.50	26.00	47.00	7.64	71.36	4.00	20.00	25.00	70.00	40.00	12.50
GPT-4-Turbo	**15.50**	**67.50**	**77.00**	14.38	72.83	**68.00**	**88.00**	**80.00**	**95.00**	**95.00**	16.00
Open-sourced Models & Agents											
Llama2-Chat-7B	0.50	2.00	0.00	0.83	0.23	0.00	0.00	0.00	0.00	0.00	1.50
Llama2-Chat-13B	1.00	3.50	0.00	0.83	0.10	0.00	0.00	0.00	0.00	0.00	1.50
AgentLM-7B	36.50	71.00	**4.00**	1.63	0.49	**12.00**	4.00	0.00	**5.00**	15.00	5.00
AgentLM-13B	39.50	**73.00**	0.00	2.75	0.45	8.00	0.00	**10.00**	**5.00**	5.00	3.00
AgentLM-70B	**49.50**	67.00	**4.00**	**10.68**	**0.66**	8.00	4.00	0.00	0.00	**40.00**	**7.50**
Ours											
BC_{base}	66.50	77.50	44.00	26.42	69.31	**12.00**	12.00	25.00	5.00	45.00	8.00
BC_{large}	73.50	83.00	60.00	**74.47**	74.19	**12.00**	**36.00**	**45.00**	5.00	65.00	8.50
AGENTEVOL	**76.50**	**88.00**	**64.00**	38.00	**82.70**	**12.00**	12.00	25.00	**60.00**	**70.00**	**9.00**

研究背景：从“提示驱动”到“交互驱动”的智能体



在经验时代，大规模、多环境强化学习成为智能体演进的关键一步

Prompt → Interaction：

- 单轮提示解决静态问题；
- 多轮交互才能解决长程、开放式任务。

经验为王：

- 来自环境的序列经验（trajectory）> 离线静态数据
- 能不断刷新策略与技能库。

核心挑战：

- 长上下文记忆、分层规划、错误恢复、信用分配、稳定训练。

在交互时代，我们需要怎样的“交互学习”？

开放共享：

- **开源框架：**
  合作开发，协同共建。
- **灵活适配：**
  支持新环境和算法的便捷适配，减少社区二次开发门槛。

环境构建：

•多轮交互： 提供智能体学习长期规划与动态调整支持连续多轮对话和状态反馈的环境。

•真实任务： 贴近真实使用场景（网页、GUI、代码等），让策略不仅能“解题”，还能持续协作与纠错。

算法优化：

- **更长交互**：适配更长交互轮次的训练规模，使智能体能充分探索状态空间积累经验。
- **稳定训练**：避免长交互训练过程的崩溃中断，使智能体能够稳健持久地积累有效经验。

主要贡献

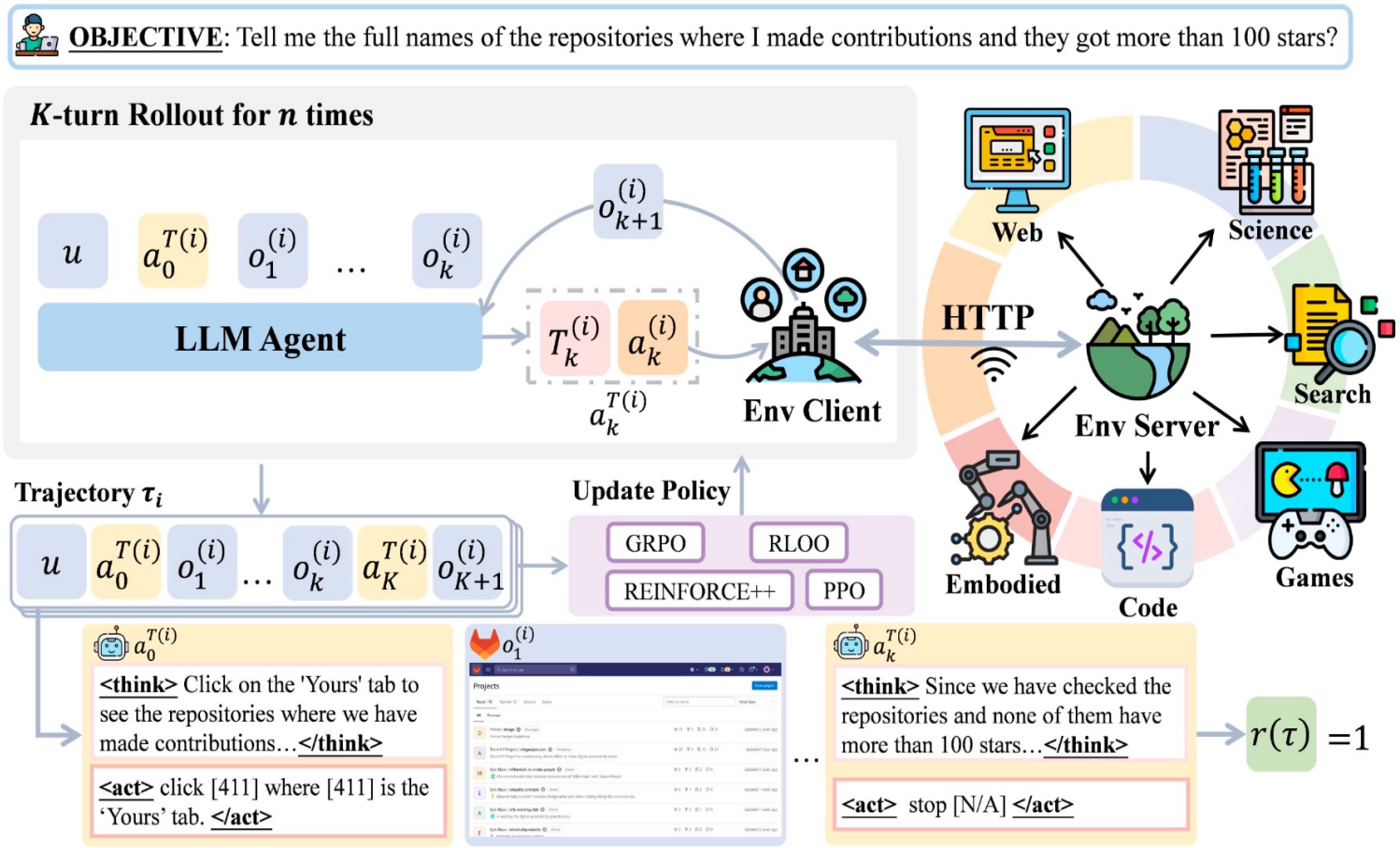
AgentGym-RL：

首个无需监督微调、具备统一端到端架构、支持交互式多轮训练，且在多类真实场景中验证有效的 LLM 智能体强化学习框架。

ScalingInter：

专为稳定长交互轮次训练、平衡探索 - 利用而设计，通过渐进式交互轮次扩展实现高效优化的强化学习算法。

AgentGym-RL：多环境、稳训练、广适配、易扩展的强化学习框架



AgentGym-RL整体架构图

多环境：

覆盖五种真实环境，满足多样化训练需求。

广适配：

同时支持在线强化学习（PPO、GRPO、RLOO、REINFORCE++）和静态优化方法（SFT、DPO、拒绝采样）两大核心训练路径。

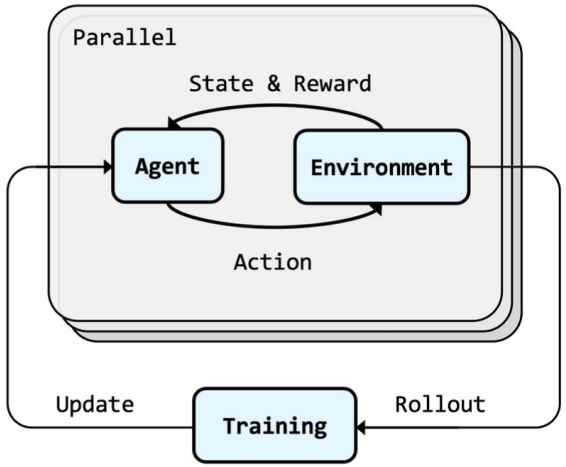
稳训练：

通过工程优化设计，确保高并发、长交互场景下的稳定训练。

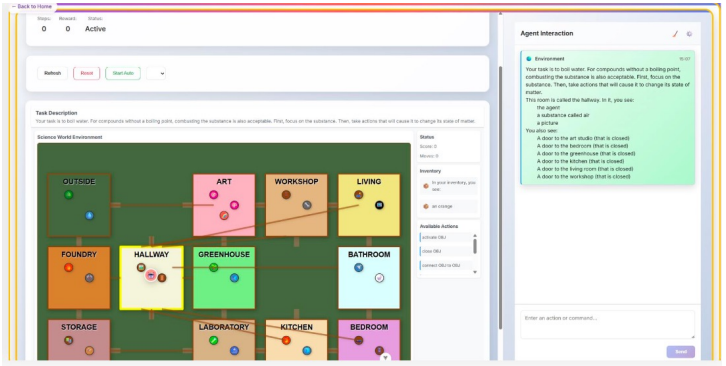
易扩展：

完善的开源生态系统与模块化架构便于二次开发，统一接口设计降低适配成本。

AgentGym-RL：环境–智能体–训练三模块



AgentGym-RL 理论模型



AgentGym-RL 可视化界面

三大核心模块

- **环境模块**：通过标准 Server–Client 架构与统一 HTTP 协议提供多种环境。
- **智能体模块**：封装智能体多轮交互中的推理与决策过程。
- **训练模块**：实现强化学习训练流程，收集轨迹并更新智能体策略。

主要工程优化

可靠性

- 优化资源管理，保证大规模多轮交互训练的稳定性。

可规模化

- 提升并行处理能力
- 支持更长训练周期

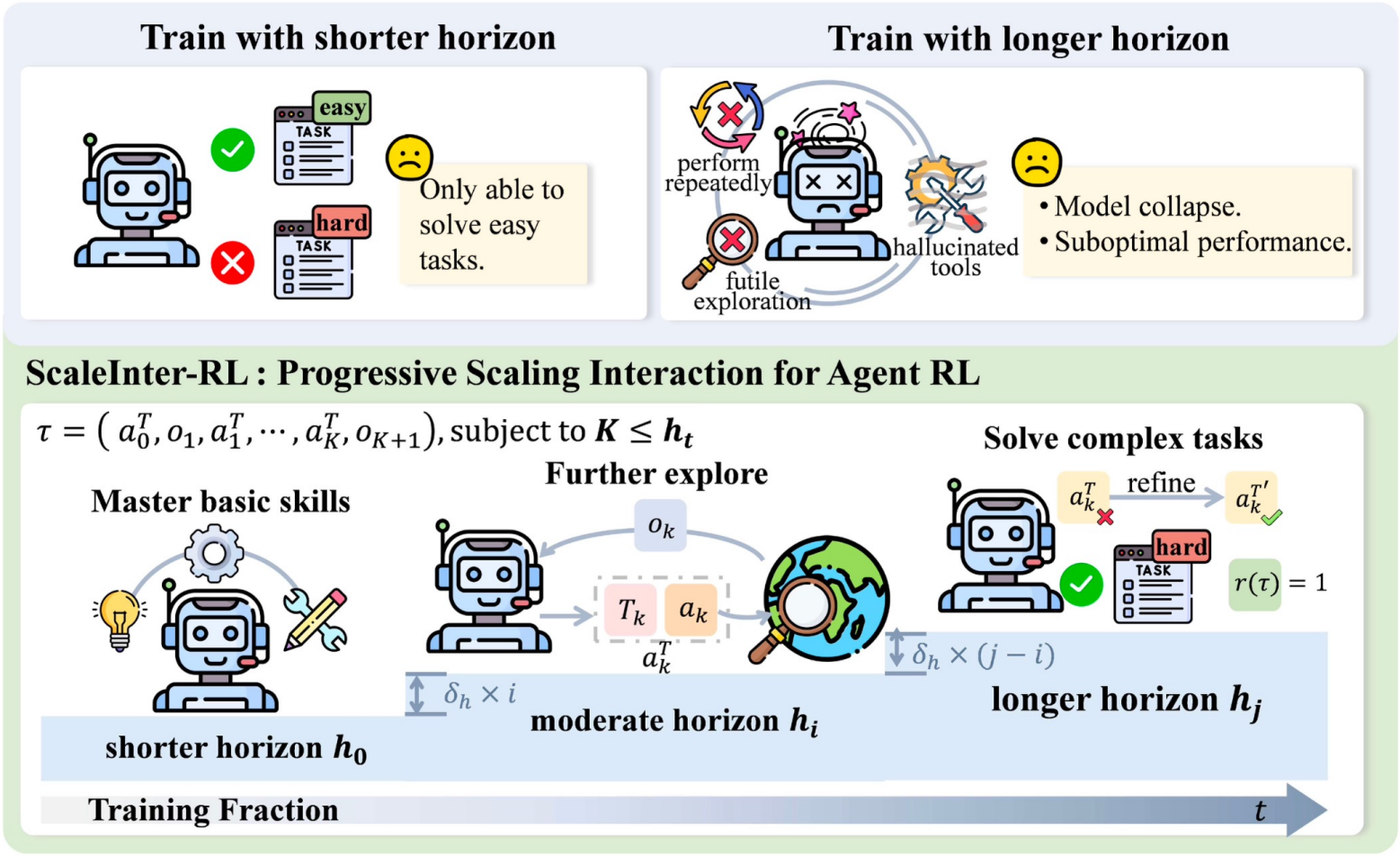
可扩展性

- 模块化、解耦式设计，核心组件可插拔。
- 支持灵活替换奖励函数、采样方法、RL算法、拓展环境，便于快速实验与复现。

社区易用性

- 提供标准化 API、可复现训练与评测流水线，降低研究门槛。
- 可视化交互界面，支持行为轨迹回放、推理过程分析，加速调试与迭代。

ScalingInter：渐进式交互轮次扩展实现探索-利用平衡



ScalingInter 算法理论图

传统固定交互轮次

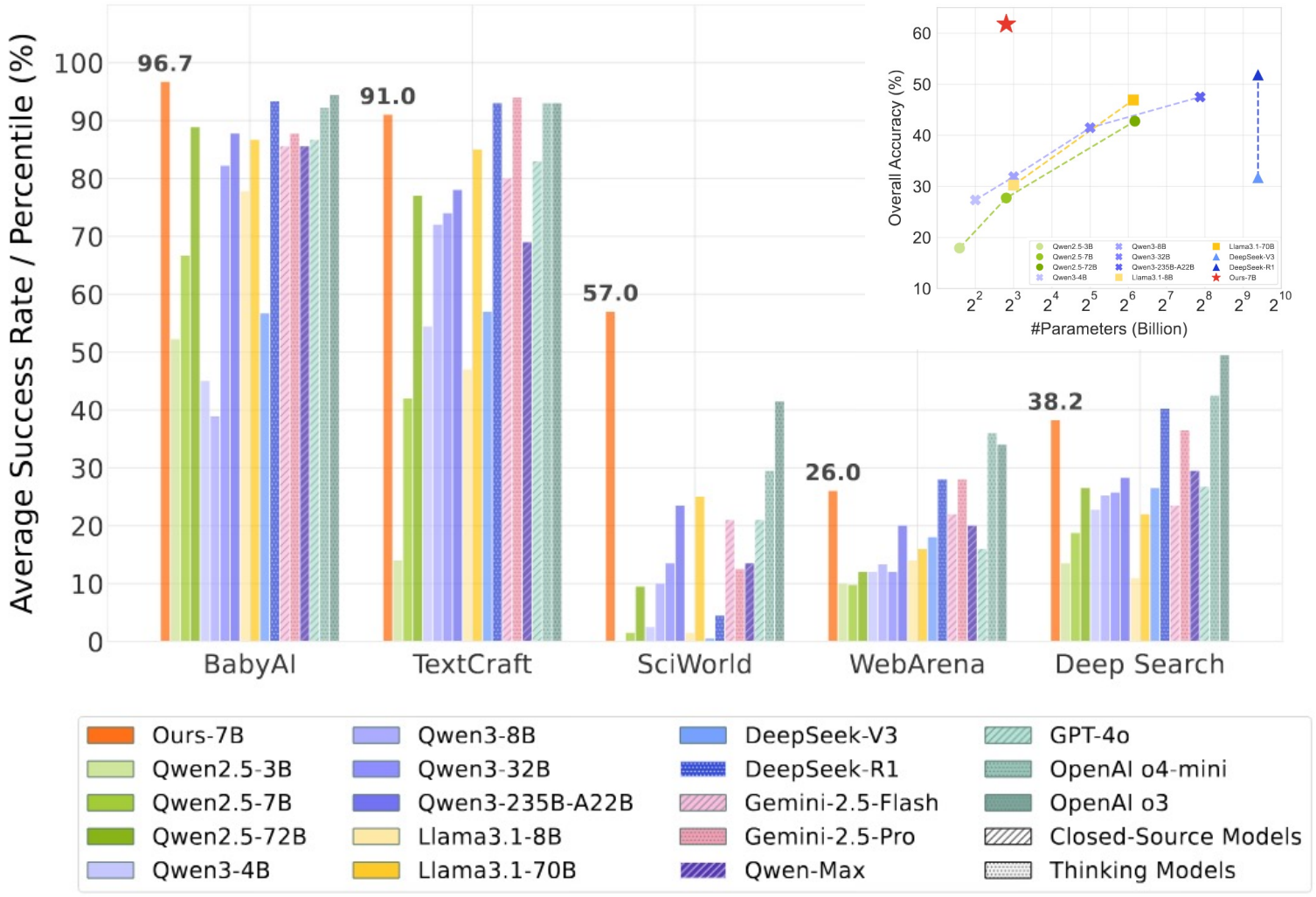
- 固定短交互：训练稳定、利用效率高，但探索受限、性能较差。
- 固定长交互：探索更丰富，但训练过程不稳定，容易崩溃。

=> 如何兼顾两者的优点?

ScalingInter：渐进式扩展交互轮数

从短交互回合开始训练，确保初期高效掌握基础技能；随训练进展逐步增加交互轮数，扩大探索深度，形成渐进式扩展，实现稳定学习到多轮交互、利用与探索的平衡。

实验结果



AgentGym-RL总体性能图

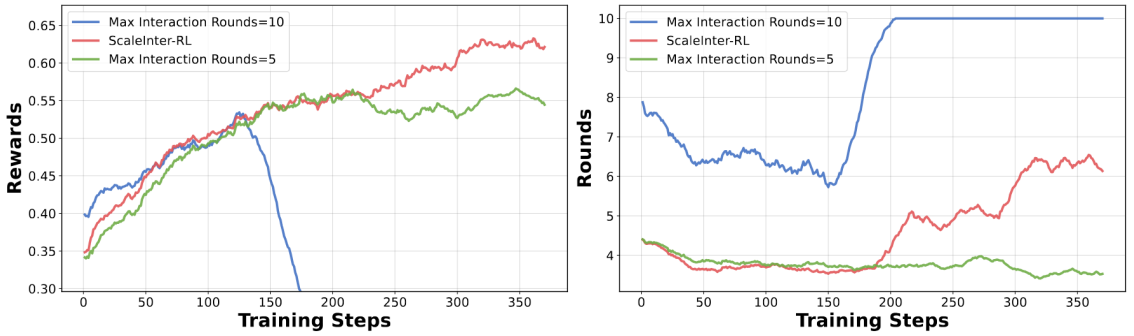
ScalingInter 性能总览

- 平均成功率达到约 58.6%，超越参数规模更大的开源模型如 Llama3.1-70B（~47%）和 Qwen2.5-72B（~43%）
- 在 WebArena 任务上，性能提升达 10%；在 TextCraft 任务上，性能提升达 30 %；在 Sciworld 任务上，性能提升近 50%，超越所有模型。

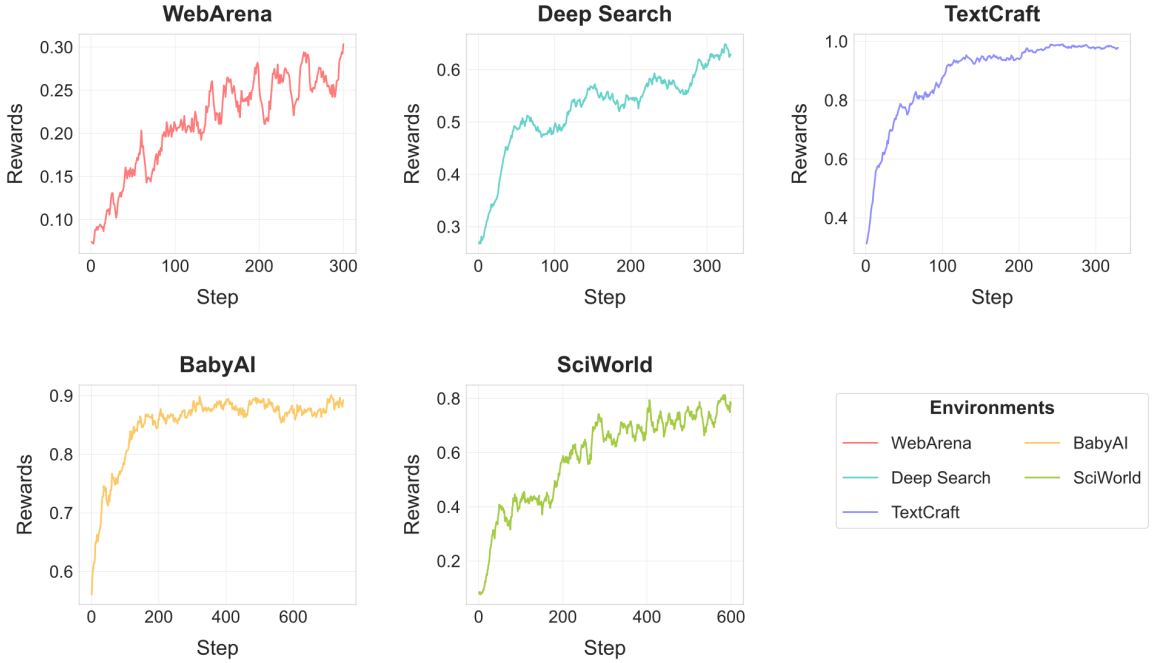
=> 进行渐进式交互规模调节的强化学习训练，效果优于单纯的模型参数规模扩大。

实验结果

Deep Search 环境中不同最长交互轮数下的训练动态



ScalingInter算法在不同环境下的训练奖励曲线

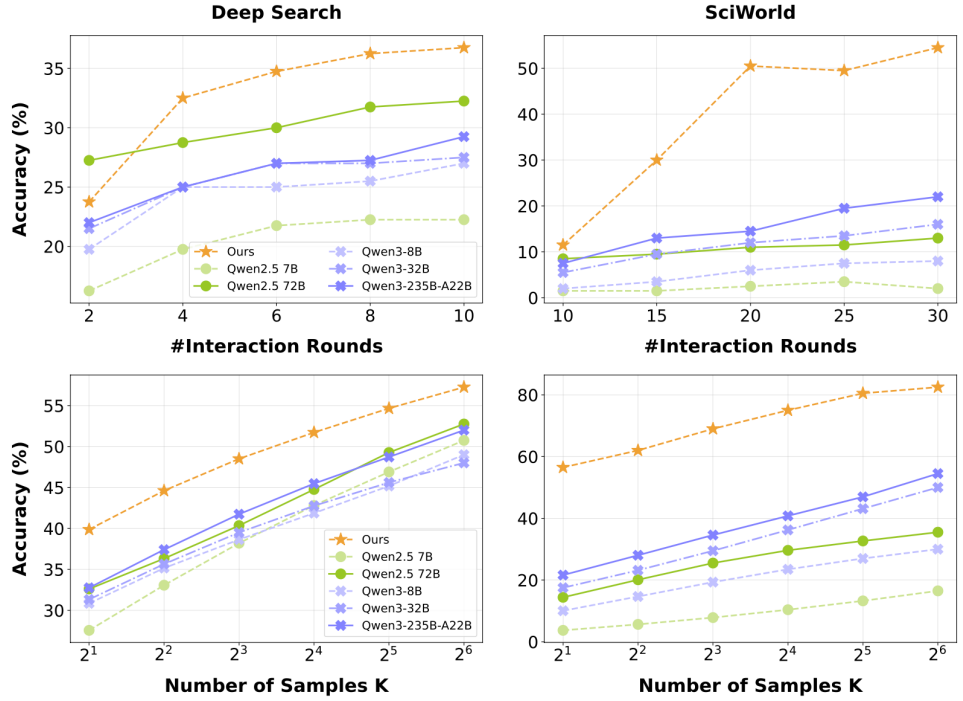


ScalingInter 提升训练稳定性

- 短周期强化学习虽能让训练保持稳定，但会限制最终性能；长周期强化学习在训练初期性能提升最快，但会在训练后期出现优化崩溃的情况。
- ScalingInter 能够平衡探索与利用的关系，规避上述两种训练方式的缺陷，最终获得更高的长期奖励。
- 在不同环境中的实验表明，ScalingInter 在不同环境下均能取得稳定、持续且客观的奖励提升。

实验结果

扩展最大交互轮数、并行采样数量时不同模型的表现



GRPO 与 REINFORCE++ 算法的比较

RL Algorithms	TextCraft	BabyAI	SearchQA
Qwen2.5-3B-Instruct			
GRPO	**75.00**	**93.33**	**25.75**
REINFORCE++	28.00	70.00	13.25
Qwen2.5-7B-Instruct			
GRPO	**83.00**	**92.22**	**34.00**
REINFORCE++	73.00	84.44	24.00

扩展测试时计算可显著提升任务准确率

- **扩展最大交互轮次可显著提升任务准确率。**
  - 智能体必须充分探索环境，才能形成稳定可复用的交互策略与行为模式。
- **增加采样数量可显著提升任务准确率。**
  - 即便在采样预算较少的情况下，我们的模型仍优于基准模型。
  - 随着采样数量增加，其性能优势始终保持稳定且显著。

GRPO 算法的性能始终显著优于 REINFORCE++ 算法

- REINFORCE++ 算法的学习信号来自完整回合蒙特卡洛回报，梯度方差大且对长轨迹随机结果敏感；GRPO 算法则通过动作相对价值的对比评估获得更加稳定的梯度，进而优化复杂低信号环境下的探索与功劳分配。

为什么要扩展目标?

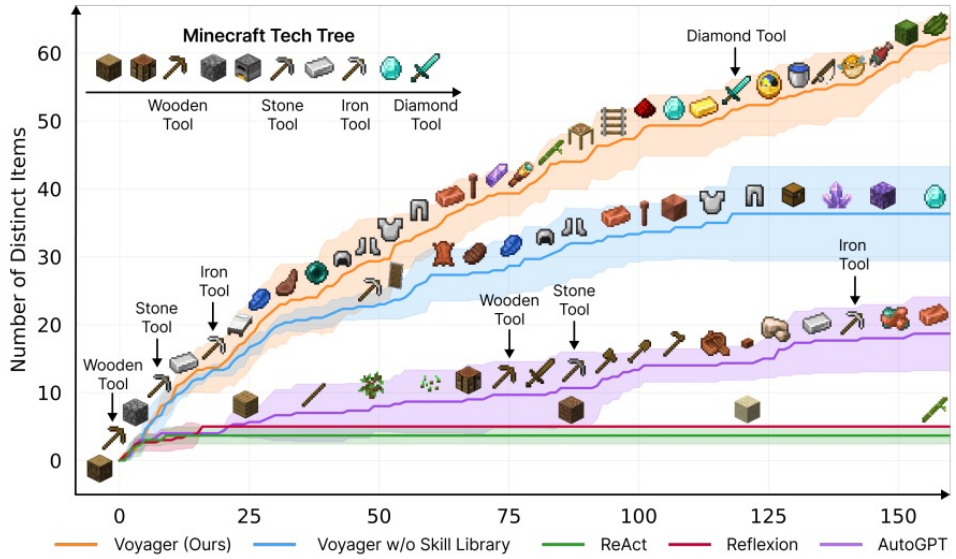
从单任务到终身任务体系

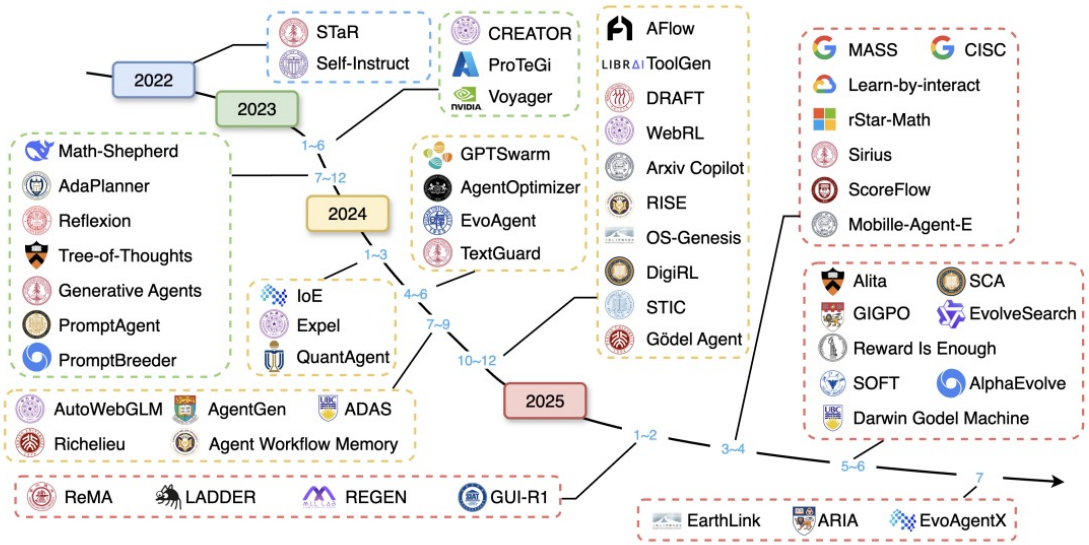
- 我们希望让智能体不再局限于完成单个任务，而是具备 **跨任务、跨环境的持续学习能力。**
- 通过构建目标体系（Goal Hierarchy），Agent 可以在不同层级重新定义目标、分解任务、规划策略。
- 扩展目标意味着智能体能够 **反思、重构、自主生成新目标**，从而具备长期成长能力。

背景研究：从自动化执行到开放式长期探索

AutoGPT / Voyager / Lifelong Learning

Voyager 展示了长期目标规划与技能积累。
AutoGPT 展示任务分解与自动执行。





[1] Auto-GPT for Online Decision Making. arXiv:2306.02224, 2023.
[2] Voyager: An Open-Ended Embodied Agent with Large Language Models. TMLR 2023.
[3] A Survey of Self-Evolving Agents: On Path to Artificial Super Intelligence. arXiv:2507.21046, 2025.

背景工作：

MineDojo → Voyager → Eureka → GR00T：目标从任务生成到奖励编程。

MineDojo（开放式环境+互联网知识）：可扩展目标与丰富交互；我们用 AgentGym 环境注册/任务库引入“环境即供给”，支撑 Environment Scaling。

Voyager（自动课程+技能库）：长期探索与技能沉淀；我们在 AgentGym-RL 做渐进式交互（ScalingInter）与技能回放，稳长链学习。

Eureka（LLM 写奖励）：目标→可执行奖励自动化；我们在 AgentGym 用奖励模板化/拒绝采样/权重重分配，降奖励工程成本。

GR00T（VLA 语言-视觉-动作）：语言目标落到多形态动作；我们以 MagicGUI 抽象触控（Tap/Scroll/Input/LongPress）+多模态目标生成，承接 Goal Scaling。

MineDojo

✨NeurIPS Outstanding Paper Award✨.

Large-scale open-ended agent learning framework in Minecraft.

Website | Paper | Code | Database | Doc | Blog | Announcement | Intro

Voyager

LLM-powered agent that masters Minecraft by in-context lifelong learning.

Website | Paper | Code | Announcement | NYTimes

[1] MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. NeurIPS 2022.
[2] Voyager: An Open-Ended Embodied Agent with Large Language Models. TMLR 2024.
[3] Eureka: Human-Level Reward Design via Coding Large Language Models. NeurIPS 2023.
[4] GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. NVIDIA Research 2025.

Self evolving思想

Goal Hierarchy + Self-Evolution

层次化目标（Goal Hierarchy）：Vision/Meta-Goal → Task → Subgoal → Skill。
高层**提出方向**，中层**拆成可执行子目标**，低层**沉淀可复用技能。**

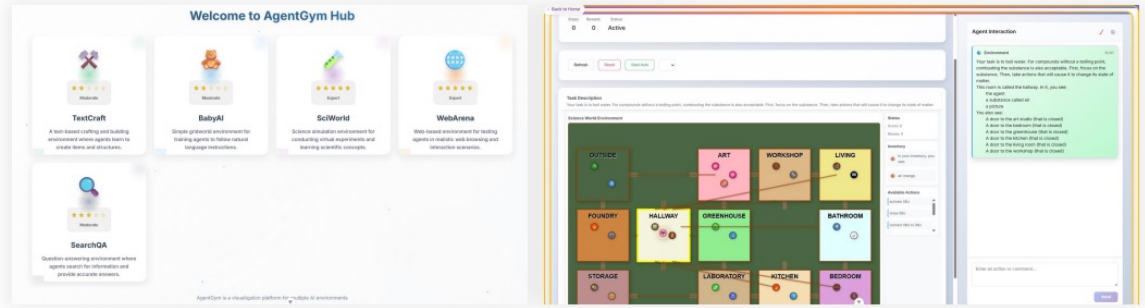
自我进化循环（Self-Evolution Loop）：**Goal → Action → Eval → Refine → New Goal。**
失败样本进入**拒绝池**，成功轨迹进入**回放 / 蒸馏**；目标权重**按价值重分配。**

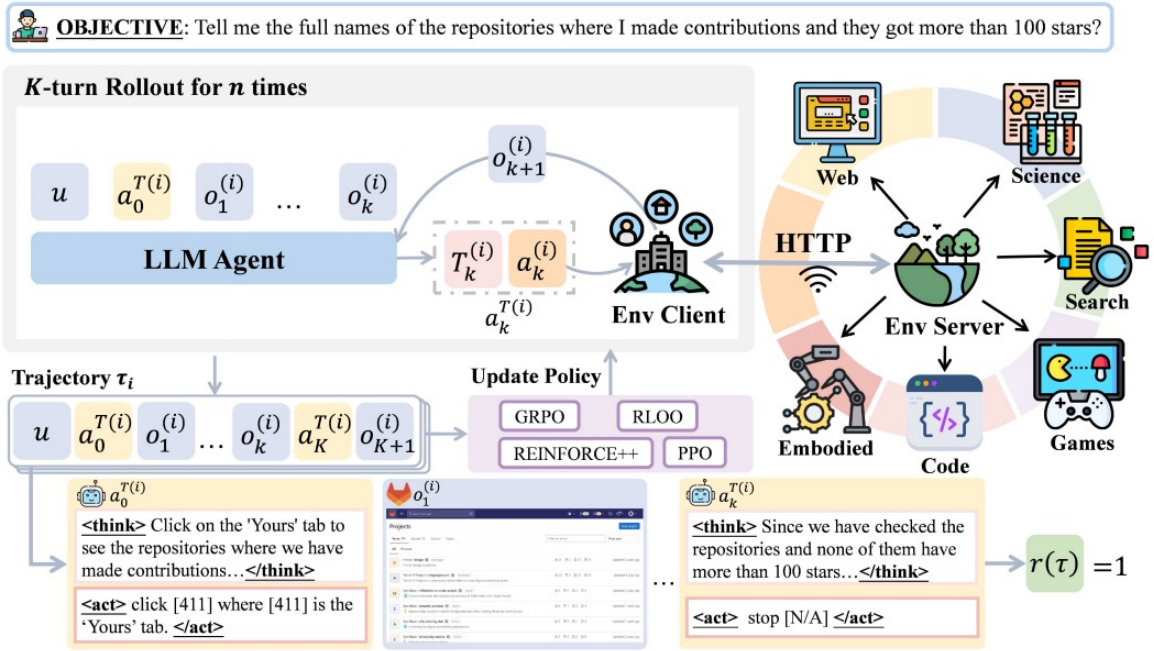
我们的实例化：
- **AgentGym**：目标重采样 + 奖励重权 → **目标可塑化**；
- **AgentGym-RL**：**渐进式交互（ScalingInter）** + 稳定训练 → **长程信用分配**；
- **MagicGUI**：**UI 状态 → 中间目标** → 触控动作序列 → **多模态目标生成。**

[1] A Survey of Self-Evolving Agents: On Path to Artificial Super Intelligence. ArXiv 2507.21046, 2025.

AgentGym-rl 实践

通过任务重采样与奖励重权，实现目标自动演化。

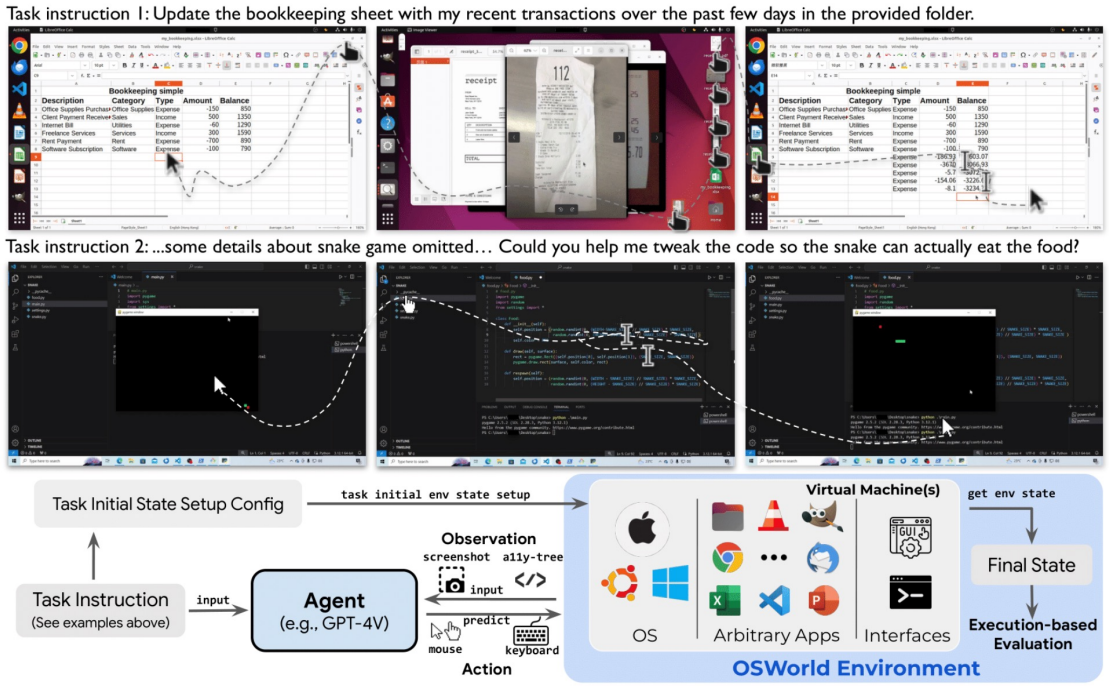




从文本到多模态环境

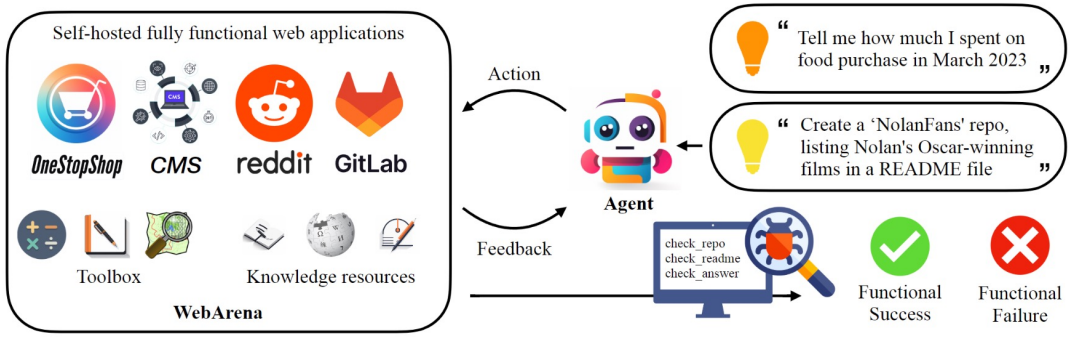
OSWorld：真实电脑多模态环境基准

- 跨 Ubuntu / Windows / macOS 操作系统
- 包含 369 个真实任务：应用操作、网页交互、文件系统 I/O、跨应用流程
- 配有初始状态 + 自动验证脚本，确保评测可重复
- 人类完成率 ~72.4%，最优模型仅 ~12.2%，主要瓶颈在 GUI grounding 与操作知识
- OSWorld-Human 版本进一步分析**效率差距**：模型行动步数常为人类的 1.4–2.7 倍



WebArena：真实可执行的网页环境基准

- 支持自托管部署，复现性强
- 模拟四大网页场景（电商 / 社区 / 协作 / CMS） + 嵌入工具 / 知识库
- 从自然语言指令 → 网页动作序列 → 验证自动评估
- GPT-4 基线约 14% 成功率，人类 ~78%，仍有巨大差距



[1] Zhou et al., WebArena: A Realistic Web Environment for Building Autonomous Agents, arXiv 2023.
[2] Xie et al., OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks. NeurIPS' 24.

从文本到多模态环境： GUI Environment

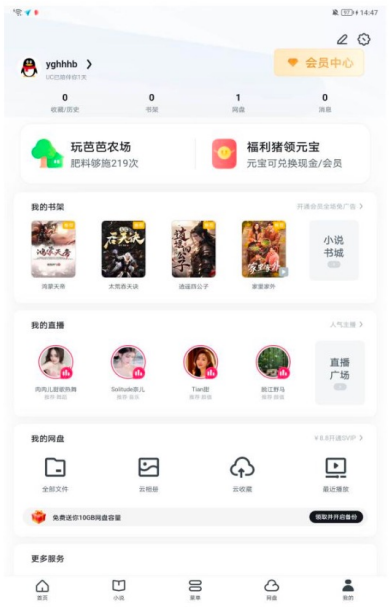
动机

- 多模态大模型推动 GUI 自动化智能体，但传统模块化混合方案对专家规则与运行环境敏感，扩展性差；端到端智能体更具可扩展潜力。

挑战

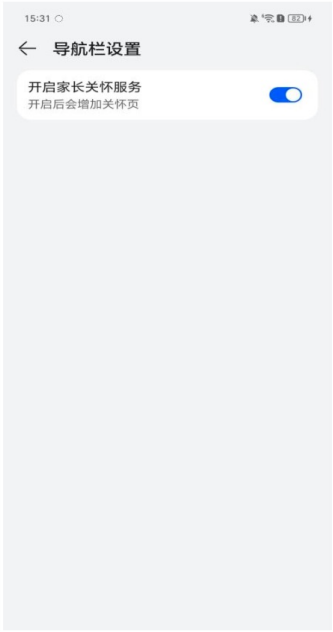
- ①数据规模与质量；②感知与定位在不同 UI 上的稳定性；③推理泛化与多步决策。

Foldable phone & Night mode



Pad & Day mode

Non-interactive



Completed

Loading

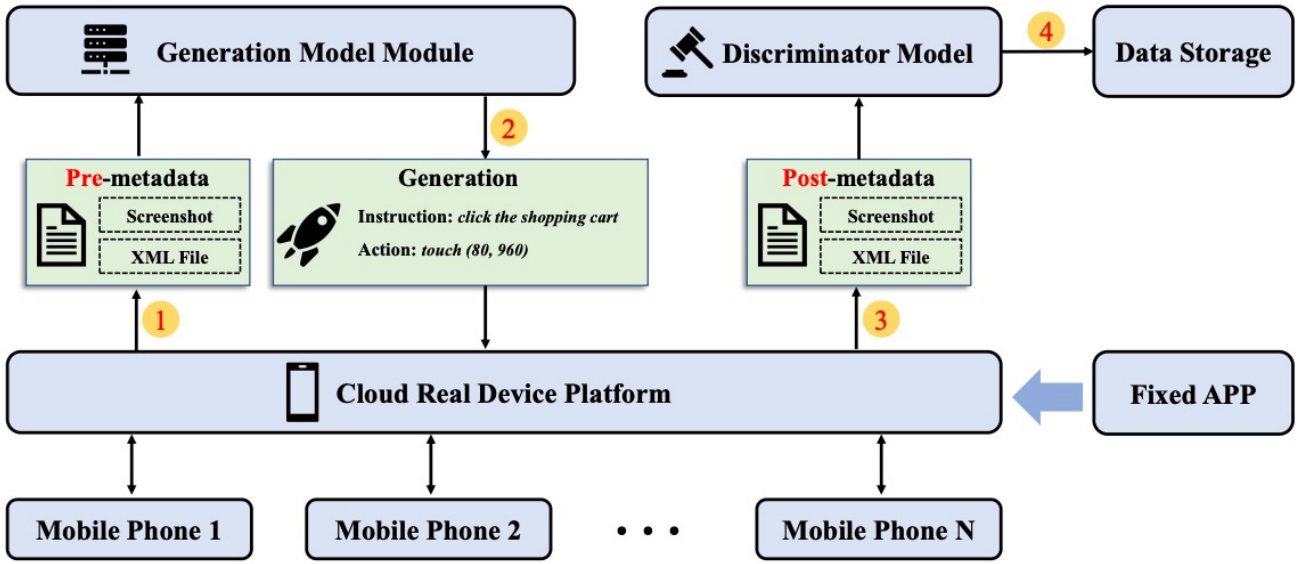
MagicGUI 环境构建

数据来源与采集方式

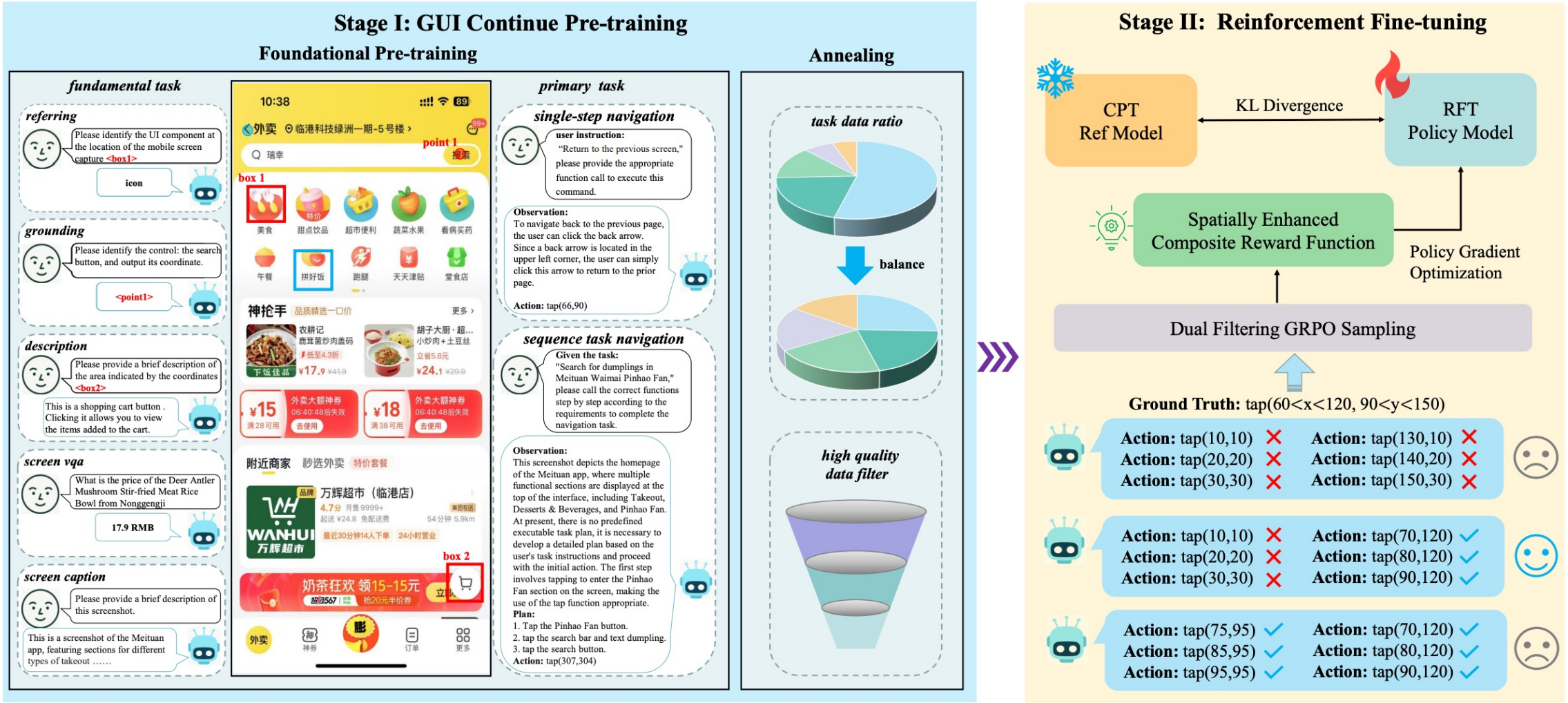
- 开源数据 + 自动抓取 + 人工采集，覆盖UI元素与导航序列。
- Cloud Real Device Platform 实机平台：多手机并行、任务下发/复位、四步工作流（预元数据→指令/动作生成→后元数据→筛选入库）。

统一动作空间

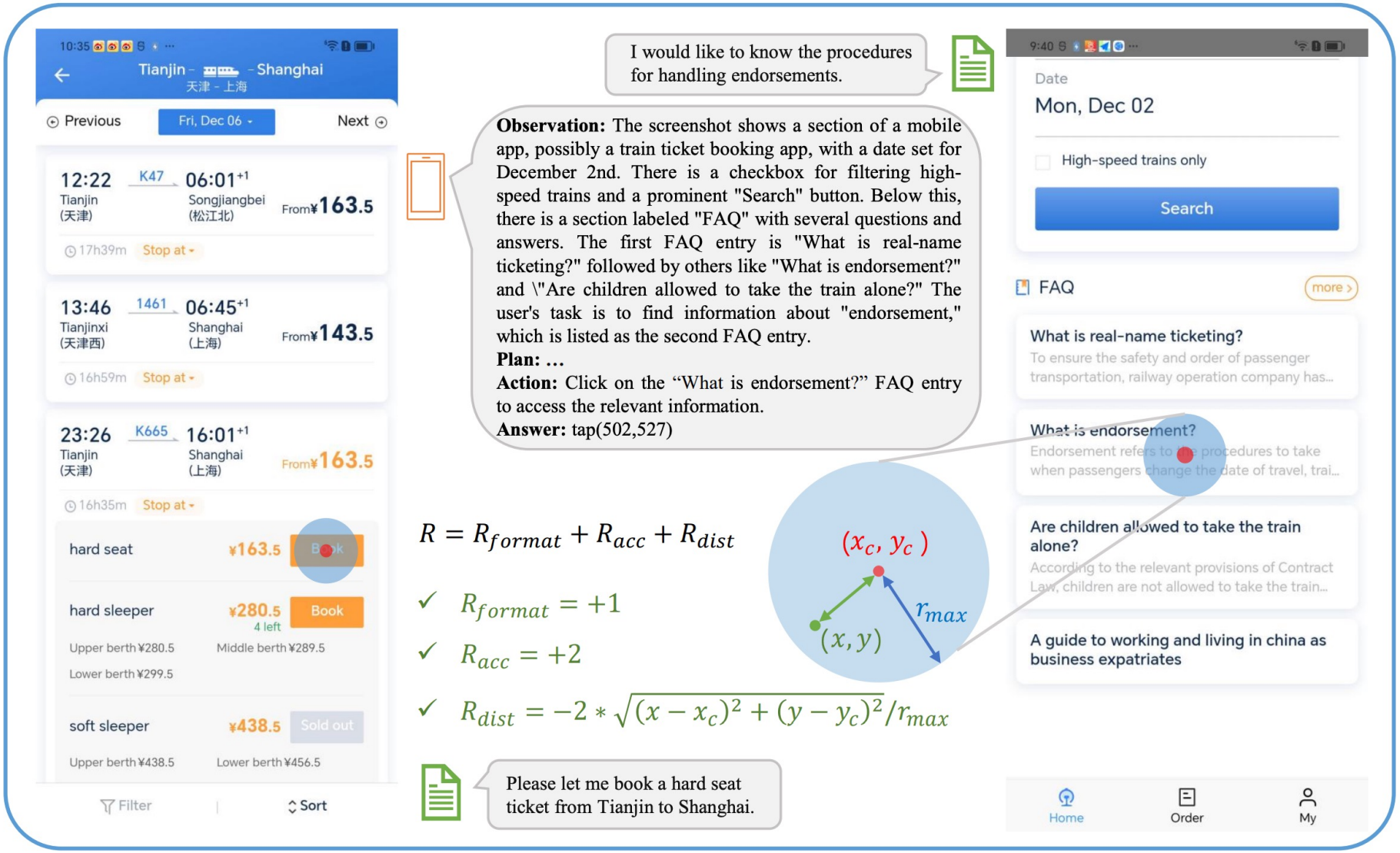
- 基本：点击、滚动、文本输入、导航返回、导航主页、长按、完成
- 扩展：等待、进入、接管、拖动、调用 API、截图、长截图和无响应动作



MagicGUI 的训练



空间增强的强化微调


I would like to know the procedures for handling endorsements.
Observation: The screenshot shows a section of a mobile app, possibly a train ticket booking app, with a date set for December 2nd. There is a checkbox for filtering high-speed trains and a prominent "Search" button. Below this, there is a section labeled "FAQ" with several questions and answers. The first FAQ entry is "What is real-name ticketing?" followed by others like "What is endorsement?" and \"Are children allowed to take the train alone?" The user's task is to find information about "endorsement," which is listed as the second FAQ entry.
Plan: …
Action: Click on the "What is endorsement?" FAQ entry to access the relevant information.
Answer: tap(502,527)
$R = R_{format} + R_{acc} + R_{dist}$
✓ $R_{format} = +1$
✓ $R_{acc} = +2$
✓ $R_{dist} = -2 * \sqrt{(x - x_c)^2 + (y - y_c)^2}/r_{max}$
(x_c, y_c)
r_{max}
(x, y)
Please let me book a hard seat ticket from Tianjin to Shanghai.
Tianjin - Shanghai
天津 - 上海
Previous
Fri, Dec 06
Next
12:22 K47 06:01+1
Tianjin (天津)
Songjiangbei (松江北)
From¥163.5
17h39m Stop at
13:46 1461 06:45+1
Tianjinxi (天津西)
Shanghai (上海)
From¥143.5
16h59m Stop at
23:26 K665 16:01+1
Tianjin (天津)
Shanghai (上海)
From¥163.5
16h35m Stop at
hard seat ¥163.5 Book
hard sleeper ¥280.5 4 left Book
Upper berth ¥280.5
Middle berth ¥289.5
Lower berth ¥299.5
soft sleeper ¥438.5 Sold out
Upper berth ¥438.5
Lower berth ¥456.5
Filter
Sort
Date
Mon, Dec 02
High-speed trains only
Search
FAQ
more
What is real-name ticketing?
To ensure the safety and order of passenger transportation, railway operation company has...
What is endorsement?
Endorsement refers to the procedures to take when passengers change the date of travel, trai...
Are children allowed to take the train alone?
According to the relevant provisions of Contract Law, children are not allowed to take the train...
A guide to working and living in china as business expatriates
Home
Order
My

MagicGUI 的目标生成

Task:
Please help me find the highest-rated hotel on Douyin's group buying page, and check the price of the comfortable king bed room.

Action History: $a_{0:t-1}$

Previous Thought: t_{t-1}

Current Screenshot: o_t

Observation:
The screenshot shows the home screen of a smartphone with various app icons visible. Currently, there is no available plan. I need to formulate a plan based on the user's task and proceed with the first step. Since I aim to open the app directly, using the API call is appropriate for accuracy.

Plan:
1. Open the Douyin app.
2. Navigate to the group buying page.
2. Navigate to the hotel group buying section.
3. Browse the list of hotels and check the ratings for each hotel.
4. Click on the highest rating hotel.
5. Click on the comfortable king bed room and check its price.
6. Task completed.

Action: Open the Douyin app via API call.

Answer: call_api(Douyin,open)

Observation:
The screenshot shows the Douyin app interface. At the top of the screen, there is a navigation bar with options such as Featured (推荐), Group Buying (团购), Nanjing (南京), Following (关注), Mall (商城), and Recommended (推荐). According to the plan, the next logical action is to switch to the Group Buying section.

Plan:
1. Open the Douyin app. -Done
2. Navigate to the group buying page.
2. Navigate to the hotel group buying section.
3. Browse the list of hotels and check the ratings for each hotel.
4. Click on the highest rating hotel.
5. Click on the comfortable king bed room and check its price.
6. Task completed.

Action: Click on the Group Buying section.

Answer: tap(308,69)

Observation:
The current screenshot shows the Douyin app's group purchase page. The interface displays various categories. Among these options, the Hotels and Homestays (酒店民宿) category is visible, which is likely the next step for accessing hotel listings. The plan aligns with user's task.

Plan:
1. Open the Douyin app. -Done
2. Navigate to the group buying page. -Done
2. Tap on the Hotels and Homestays section. **Plan updated**
3. Browse the list of hotels and check the ratings for each hotel.
4. Click on the highest rating hotel.
5. Click on the comfortable king bed room and check its price.
6. Task completed.

Action: Tap on the Hotels and Homestays section.

Answer: tap(693,202)

Observation:
The current screenshot shows a list of hotels and homestays available for booking in Nanjing. The interface includes various options with ratings, distances, and prices. To identify the hotel with the highest rating, sorting the list by rating would be helpful. Selecting this smart sorting (智能排序) feature may enable us to filter or sort hotels based on their ratings.

Plan updated: Shortcut identified

Plan:
1. Open the Douyin app. -Done
2. Navigate to the group buying page. -Done
2. Tap on the Hotels and Homestays section. -Done
3. Sort by highest rating.
4. Click on the highest rating hotel.
5. Click on the comfortable king bed room and check its price.
6. Task completed.

Action: Tap on the Smart Sorting option.

Answer: tap(427,653)

Observation:
The current screenshot displays a sorting menu with options including high rating preference. To find the hotel with the highest rating, I should select the option High Rating Preference (好评优先) to sort the listings by rating, which aligns with the plan.

Plan:
1. Open the Douyin app. -Done
2. Navigate to the group buying page. -Done
2. Tap on the Hotels and Homestays section. -Done
3. Sort by highest rating.
4. Click on the highest rating hotel.
5. Click on the comfortable king bed room and check its price.
6. Task completed.

Action: Click on High Rating Preference to sort listings by rating.

Answer: tap(183,315)

Observation:
The screenshot shows a list of hotels sorted by ratings. The highest-rated hotel displayed is "Nanjing Getaway Meijia Hotel" (南京歌途·美家酒店). It has a rating of 4.9, which is higher than the other hotels listed. I need to tap on this hotel to proceed with checking the room prices for a comfortable king-size bed.

Plan:
1. Open the Douyin app. -Done
2. Navigate to the group buying page. -Done
2. Tap on the Hotels and Homestays section. -Done
3. Sort by highest rating. -Done
4. Click on the highest rating hotel.
5. Click on the comfortable king bed room and check its price.
6. Task completed.

Action: Tap on the hotel with the highest rating.

Answer: tap(460,227)

Observation:
The current screenshot displays details of "Nanjing Getaway Meijia Hotel". Specific details show that the comfort double room (舒适大床房) is priced at 89 yuan, originally 158 yuan. The user task seems completed as the information requested is displayed on the screen.

Plan updated: Early termination

Plan:
1. Open the Douyin app. -Done
2. Navigate to the group buying page. -Done
2. Tap on the Hotels and Homestays section. -Done
3. Sort by highest rating. -Done
4. Click on the highest rating hotel. -Done
5. ~~Click on the comfortable king bed room and check its price.~~
5. Task completed.

Action: End the task as the requested information is visible on the screen

Answer: finish()

从 UI 状态预测目标，形成视觉驱动的目标生成

MagicGUI 的目标生成

Autonomous task execution without knowledge injection

Task:

Please help me safely clear the cache of the Baidu app.

Observation:
The current screenshot shows the mobile home screen, comprising five functional modules: Popular Services (热门服务), Shopping (购物), Daily Videos (每日视频), Mini Games (小游戏), and Search & Discover (搜索发现). At the bottom, there are icons such as Phone, Messaging, Browser, and Camera. I need to formulate a plan based on the user's task. Since, the Baidu app is not directly accessible from this screenshot, I need to use the API call to initiate it.

Plan:
1. Open Baidu app.
2. Go to Baidu app's My Profile.
3. Click on Cache Clear button.
4. Task completed

Action:
call api(Baidu,open).

Observation:
The current screenshot shows the Baidu app's homepage. There's a search bar at the top, recommended articles in the middle, and a bottom navigation bar with options like Baidu (百度), Video (视频), AI , Messages (消息), and My Profile (我的). According to the plan, the next step is to tap on the My profile button.

Plan:
1. Open Baidu app.
2. Go to Baidu app's My Profile.
3. Click on Cache Clear button.
4. Task completed

Action:
tap(874,972)

Observation:
This is the My Profile (我的) page of the Baidu app. On this page, users have access to functionalities such as Favorites (收藏), History (历史), Bookshelf (书架), File Download (文件下载), Orders (订单), Wallet (钱包), Cloud Drive (网盘) and Cache Clear (清理缓存). According to the task plan, the subsequent step is to touch the Cache Clear button.

Plan:
1. Open Baidu app.
2. Go to Baidu app's My Profile.
3. Click on Cache Clear button.
4. Task completed

Action:
tap(673,472)

Observation:
This screenshot shows there's 67.5M of cache, and there's a Safe Clean (放心清理) button in the middle. So, the next step is to tap the Safe Clean button. Updating the plan to align with user's task.

Plan:
1. Open Baidu app.
2. Go to Baidu app's My Profile.
3. Click on Cache Clear button.
4. Click on the Safe Clean button
5. Task completed

Action:
tap(494,294)

Observation:
The current screenshot shows that the cache has been successfully cleared, and the user's task seems to be completed.

Plan:
1. Open Baidu app.
2. Go to Baidu app's My Profile.
3. Click on Cache Clear button.
4. Click on the Safe Clean button
5. Task completed

Action:
finish()

实验与结论

- 智能体能跨任务**复用策略，实现自主任务生成。**
- 目标 Scaling = 通用 Agent 的关键：**从“完成任务”到“创造任务”。**

Agent Models	Routine			Instruction			Complex			Handling
	Type	Grd	SR	Type	Grd	SR	Type	Grd	SR	Exception
Closed-source Models										
GPT-4o (Hurst et al., 2024)	49.3	16.7	4.6	56.6	13.5	19.8	49.0	14.6	7.4	85.1
Gemini 2.0 (Pichai et al., 2024)	89.2	49.4	34.7	84.1	54.2	51.4	83.3	50.3	42.0	73.7
Open-source Models										
InternVL-2-8B (Chen et al., 2024c)	30.1	2.8	1.3	37.1	4.0	15.8	17.1	6.0	1.3	70.8
Qwen2-VL-7B (Wang et al., 2024c)	71.7	41.0	28.1	73.6	43.9	41.5	65.6	28.7	21.2	68.3
Qwen2.5-VL-7B (Bai et al., 2025)	94.3	92.6	76.3	89.3	95.7	83.6	86.6	69.6	60.0	67.0
UI-TARS-7B (Qin et al., 2025)	83.5	84.9	73.3	76.6	85.6	69.8	91.4	69.1	67.0	3.6
UI-TARS-1.5-7B (Seed, 2025)	85.6	96.2	81.5	78.6	92.1	72.2	**94.7**	74.3	71.1	1.0
MiMo-VL-7B-SFT (Xiaomi, 2025)	93.0	77.9	65.3	89.7	85.7	75.4	89.1	80.1	71.0	57.0
AgentCPM-GUI (Zhang et al., 2025b)	84.3	92.2	75.1	70.4	80.7	56.0	72.3	54.6	39.4	2.4
MagicGUI-CPT	98.5	**98.5**	97.2	95.5	**96.3**	92.9	88.5	**82.3**	72.9	**93.2**
MagicGUI-RFT	**99.7**	97.5	**97.5**	**97.2**	95.6	**94.0**	92.1	80.4	**74.1**	92.1

Agent Models	AC-Low		AC-High		GUI-Odyssey	
	Type	SR	Type	SR	Type	SR
Closed-source Models						
GPT-4o (Hurst et al., 2024)	-	19.5	-	20.8	-	20.4
Gemini 2.0 (Pichai et al., 2024)	-	28.5	-	60.2	-	3.3
Claude 2.0 (Anthropic, 2024)	-	28.5	-	12.5	60.9	-
Open-source Models						
Qwen2-VL-7B (Wang et al., 2024c)	55.7	36.2	45.8	21.2	58.6	13.3
Qwen2.5-VL-7B (Bai et al., 2025)	94.1	85.0	75.1	62.9	59.5	46.3
Aguvis-7B (Xu et al., 2024)	93.9	89.4	65.6	54.2	26.7	13.5
OS-Atlas-7B (Wu et al., 2024c)	73.0	67.3	70.4	56.5	91.8*	76.8*
UI-TARS-7B (Qin et al., 2025)	95.2	91.8	81.6	74.4	86.1	67.9
AgentCPM-GUI (Zhang et al., 2025b)	94.4	90.2	77.7	69.2	**90.9**	**75.0**
MagicGUI-CPT	94.5	86.7	84.6	73.1	90.4	73.5
MagicGUI-RFT	**97.2**	**93.5**	**84.7**	**76.3**	89.7	74.3

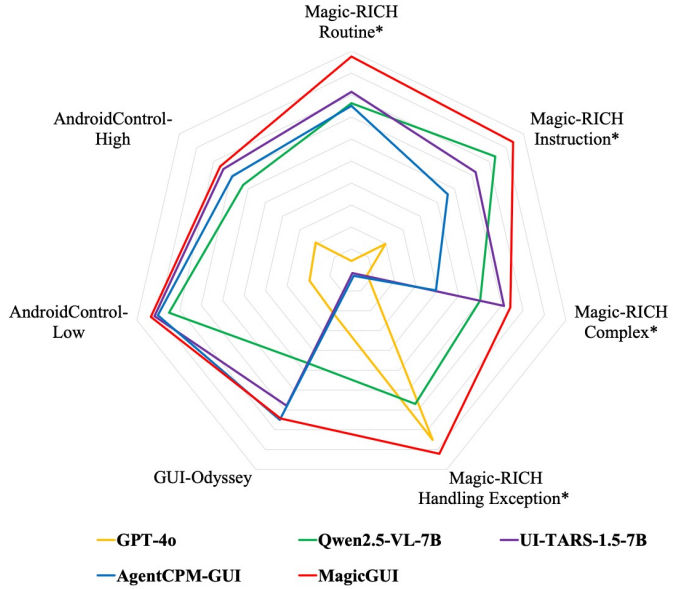
实验结果

自建基准 Magic-RICH

- Step Success Rate（SR）：Routine ≈97.5%、Instruction ≈94.0%、Complex ≈74.1%（RFT）。

公开基准与感知定位

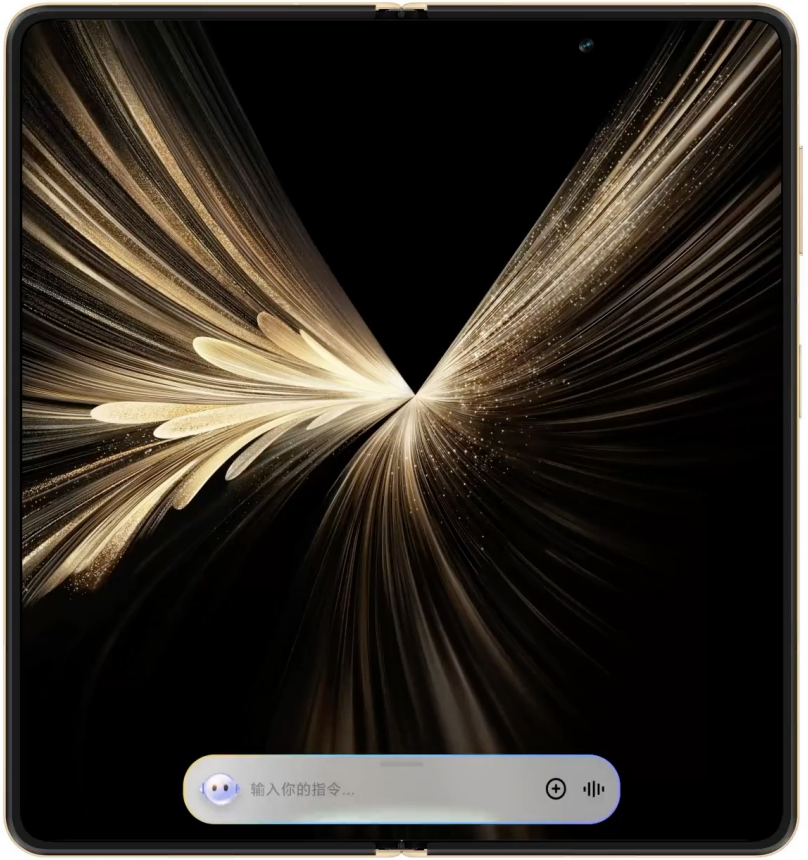
- ScreenSpot v2 mobile：90.2%；OS-Atlas-mobile：95.2%
- 与 AndroidControl / GUI-Odyssey 等多项基准比较具竞争力



Agent Models	ScreenQA-short	ScreenSpot v2 mobile	Os-Atlas-mobile
Closed-source Models			
GPT-4o (Hurst et al., 2024)	90.3	10.6	4.6
Gemini 2.0 (Pichai et al., 2024)	90.4	10.6	5.8
Open-source Models			
InternVL-2-8B (Chen et al., 2024c)	88.4	4.2	2.4
Qwen2-VL-7B (Wang et al., 2024c)	92.6	70.7	27.2
Qwen2.5-VL-7B (Bai et al., 2025)	92.1	56.1	26.6
UI-TARS-7B (Qin et al., 2025)	**95.4**	88.6	82.5
UI-TARS-1.5-7B (Seed, 2025)	93.0	85.8	79.3
MagicGUI-CPT	94.6	**90.2**	**95.2**

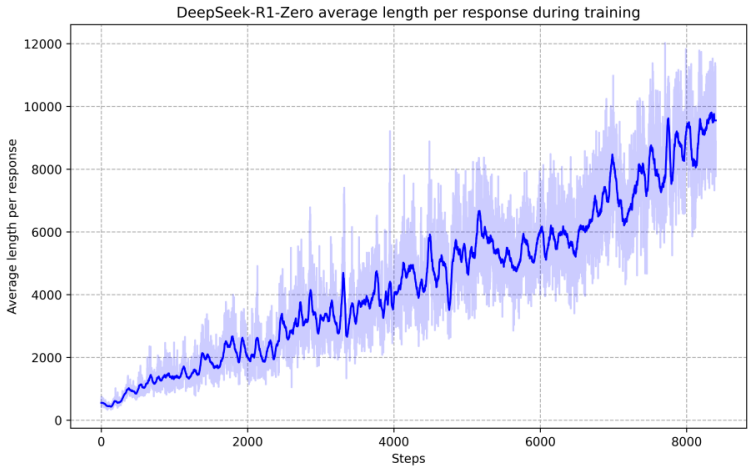
Agent Models	Routine			Instruction			Complex			Handling
	Type	Grd	SR	Type	Grd	SR	Type	Grd	SR	Exception
Closed-source Models										
GPT-4o (Hurst et al., 2024)	49.3	16.7	4.6	56.6	13.5	19.8	49.0	14.6	7.4	85.1
Gemini 2.0 (Pichai et al., 2024)	89.2	49.4	34.7	84.1	54.2	51.4	83.3	50.3	42.0	73.7
Open-source Models										
InternVL-2-8B (Chen et al., 2024c)	30.1	2.8	1.3	37.1	4.0	15.8	17.1	6.0	1.3	70.8
Qwen2-VL-7B (Wang et al., 2024c)	71.7	41.0	28.1	73.6	43.9	41.5	65.6	28.7	21.2	68.3
Qwen2.5-VL-7B (Bai et al., 2025)	94.3	92.6	76.3	89.3	95.7	83.6	86.6	69.6	60.0	67.0
UI-TARS-7B (Qin et al., 2025)	83.5	84.9	73.3	76.6	85.6	69.8	91.4	69.1	67.0	3.6
UI-TARS-1.5-7B (Seed, 2025)	85.6	96.2	81.5	78.6	92.1	72.2	**94.7**	74.3	71.1	1.0
MiMo-VL-7B-SFT (Xiaomi, 2025)	93.0	77.9	65.3	89.7	85.7	75.4	89.1	80.1	71.0	57.0
AgentCPM-GUI (Zhang et al., 2025b)	84.3	92.2	75.1	70.4	80.7	56.0	72.3	54.6	39.4	2.4
MagicGUI-CPT	98.5	**98.5**	97.2	95.5	**96.3**	92.9	88.5	**82.3**	72.9	**93.2**
MagicGUI-RFT	**99.7**	97.5	**97.5**	**97.2**	95.6	**94.0**	92.1	80.4	**74.1**	92.1

MagicGUI 的目标生成

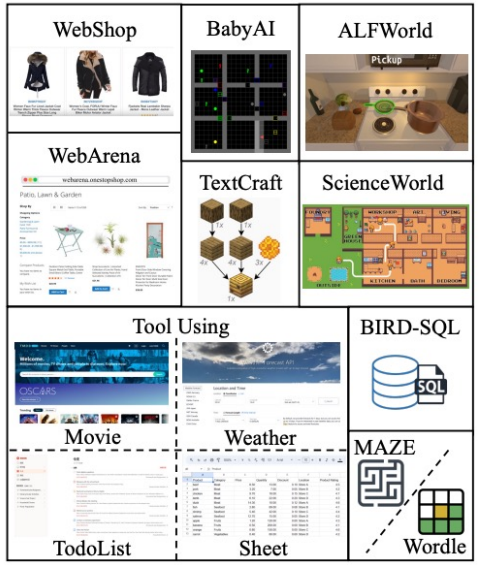


环境×交互×目标 = 自我进化的闭环

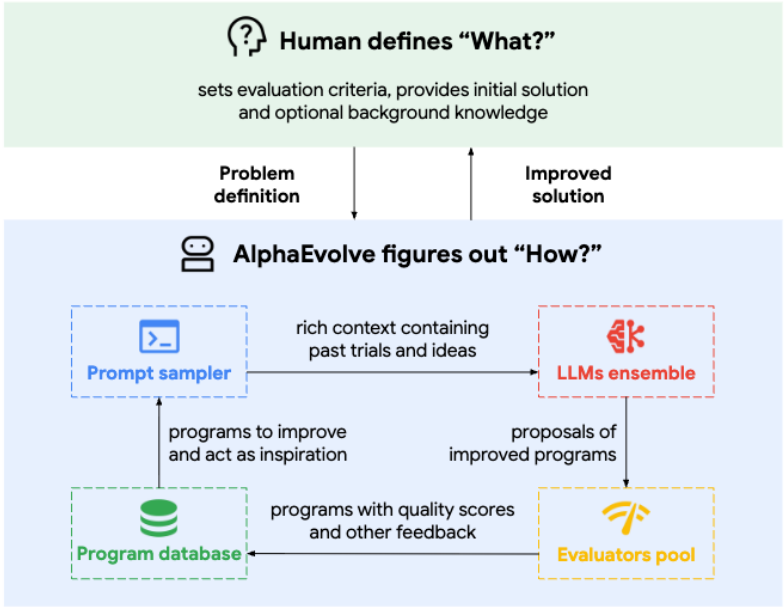
DeepSeek-R1：
面对一个 Rule-based 的奖励，
针对性地进行优化



AgentGym：
在多个序列决策任务的环境下进行探索
难以取得较好的泛化



AlphaEvolve：
在代码环境中，不断进行“提出想法-测试-留优去劣-继续提出新想法”的循环



[1] DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948, 2025.
[2] AgentGym: Evolving Large Language Model-based Agents across Diverse Environments. arXiv:2406.04151, 2024.
[3] AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery. arXiv:2506.13131, 2025.

谢谢大家！

复旦大学
黄萱菁